

Engineering a Safer and More Secure World

Nancy Leveson

MIT



Bottom Line Up Front (BLUF)

- Complexity is reaching a new level (tipping point)
  - Old approaches becoming less effective
  - New causes of mishaps appearing (especially related to use of software and autonomy)
- Traditional approaches do not provide the information necessary to prevent losses in these systems
- Need a paradigm change

Change focus

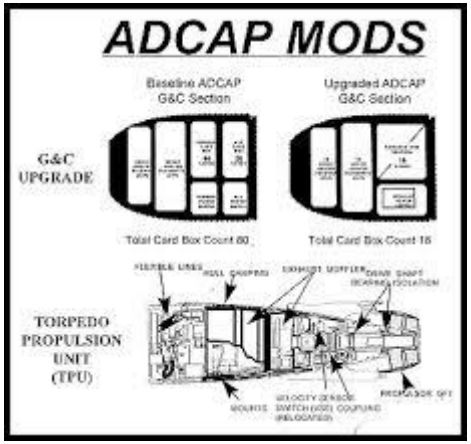
~~Increase component reliability (analytic decomposition)~~

Enforce safe behavior (dynamic control using systems theory)

BLUF (2)

- Allows creation of new analysis and engineering approaches
  - More powerful and inclusive
  - Orders of magnitude less expensive
  - Work on very complex systems (top-down system engineering)
  - Design safety and security and other properties in from the beginning
  - Compliant with MIL-STD-882E and other military standards

- New paradigm works better than old techniques:
  - Empirical evaluations and controlled studies show it finds more causal scenarios (the “unknown unknowns”)
  - Can be used before a detailed design exists to create safety and security requirements


ADCAP MODS
Baseline ADCAP
G&C Section
Upgraded ADCAP
G&C Section
G&C
UPGRADE
Total Card Box Count 60
Total Card Box Count 16
TORPEDO
PROPULSION
UNIT
(TPU)

System Safety

C.O. Miller

- Emphasizes building in safety rather than adding it on to a completed design
- Looks at systems as a whole, not just components
  - A top-down systems approach to accident prevention
- Takes a larger view of accident causes than just component failures (including interactions among components and management)
- Emphasizes hazard analysis and design to eliminate or control hazards
- Emphasizes qualitative rather than quantitative approaches

System Safety Overview

- A planned, disciplined, and systematic approach to preventing or reducing accidents throughout the life cycle of a system.
- "Organized common sense" (Mueller, 1968)
- Primary concern is the management of hazards

Hazard	**Through**
identification	analysis
elimination	design
control	management

- MIL-STD-882

(Atlas)

Goal for Session: Answer the Following Questions:

- Why do we need something new?
- What is STAMP and how does it differ from what people do now?
- What kinds of tools are available?
- How is it being used?
- Does it work?

Why do we need something new?

Our current tools are all 40-65 years old but our technology is very different today

1940 1950 1960 1970 1980 1990 2000 2010 2020

FMEA

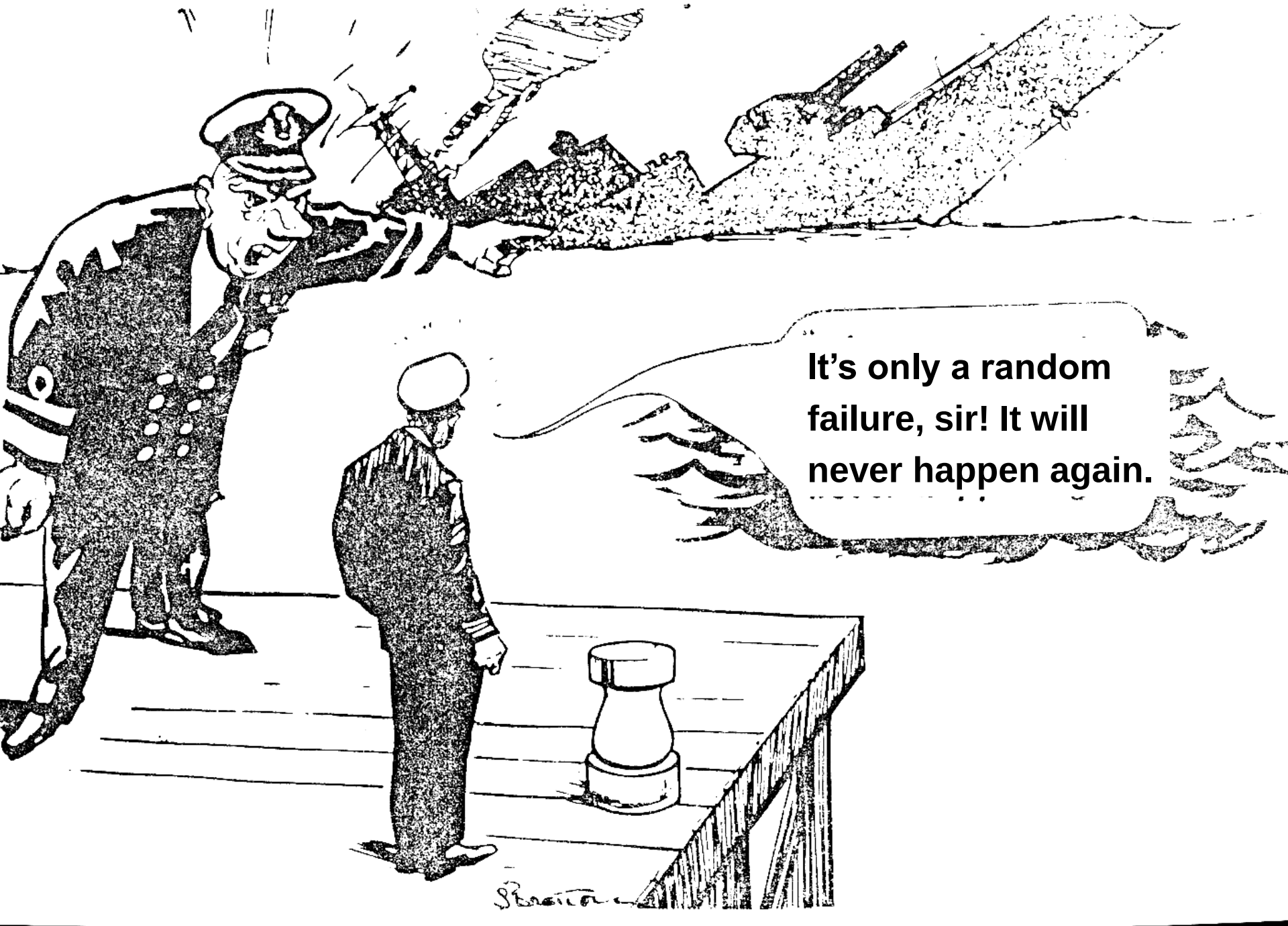
FTA HAZOP

ETA

Bow Tie (CCA) FTA + ETA

- Introduction of computer control
- Exponential increases in complexity
- New technology
- Changes in human roles

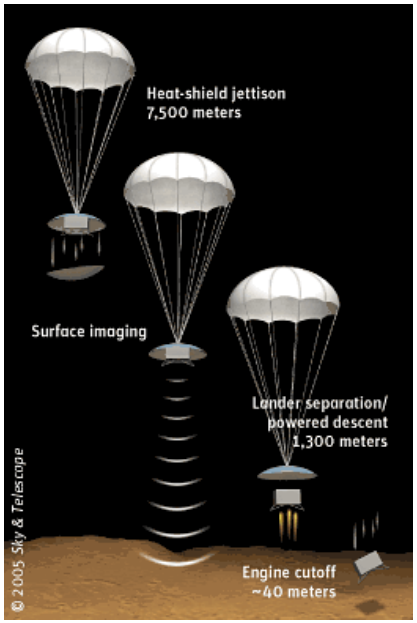
Assumes accidents caused by component failures


It's only a random failure, sir! It will never happen again.

What Failed Here?

- Navy aircraft were ferrying missiles from one location to another.
- One pilot executed a planned test by aiming at aircraft in front and firing a dummy missile.
- Nobody involved knew that the software was designed to substitute a different missile if the one that was commanded to be fired was not in a good position.
- In this case, there was an antenna between the dummy missile and the target so the software decided to fire a live missile located in a different (better) position instead.

Accident with No Component Failures



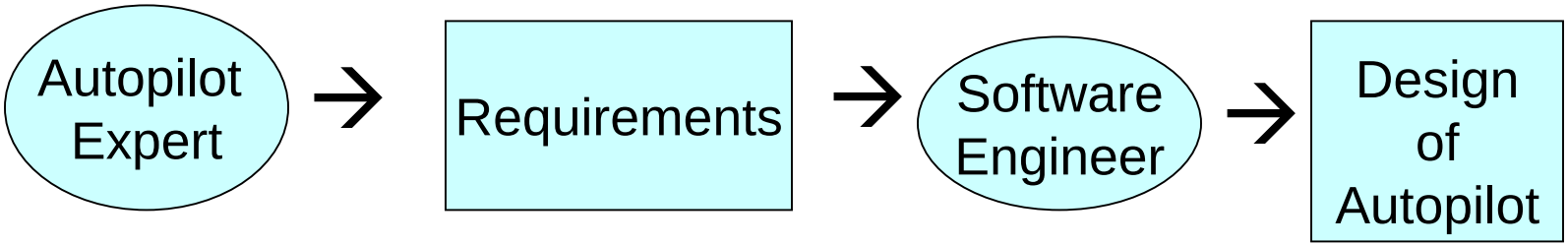
- Mars Polar Lander
  - Have to slow down spacecraft to land safely
  - Use Martian atmosphere, parachute, descent engines (controlled by software)
  - Software knows landed because of sensitive sensors on landing legs. Cut off engines when determine have landed.
  - But “noise” (false signals) by sensors generated when landing legs extended. Not in software requirements.
  - Software not supposed to be operating at that time but software engineers decided to start early to even out the load on processor
  - Software thought spacecraft had landed and shut down descent engines while still 40 meters above surface

Two Types of Accidents

- **Component Failure Accidents**
  - Single or multiple component failures
  - Usually assume random failure
- **Component Interaction Accidents**
  - Arise in interactions among components
  - Related to complexity (coupling) in our system designs, which leads to system design and system engineering errors
  - No components may have “failed”
  - Exacerbated by introduction of computers and software but the problem is system design errors

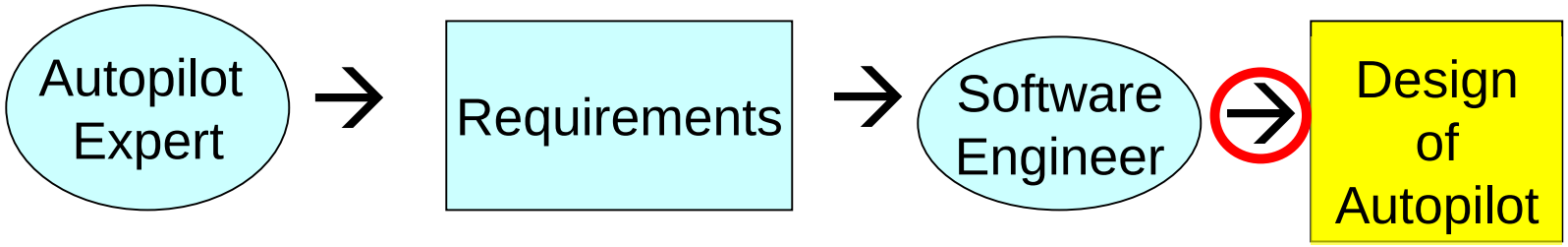
The role of software in accidents almost always involves flawed requirements

- Incomplete or wrong assumptions about operation of controlled system or required operation of computer
- Unhandled controlled-system states and environmental conditions



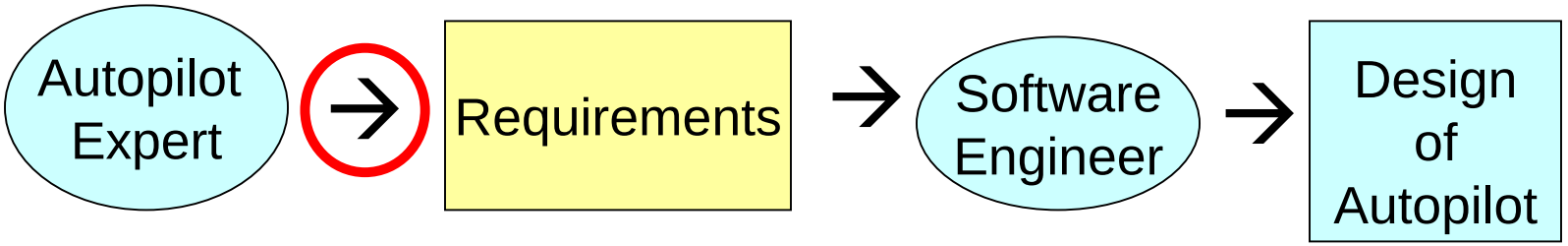
The role of software in accidents almost always involves flawed requirements

- Incomplete or wrong assumptions about operation of controlled system or required operation of computer
- Unhandled controlled-system states and environmental conditions



The role of software in accidents almost always involves flawed requirements

- Incomplete or wrong assumptions about operation of controlled system or required operation of computer
- Unhandled controlled-system states and environmental conditions



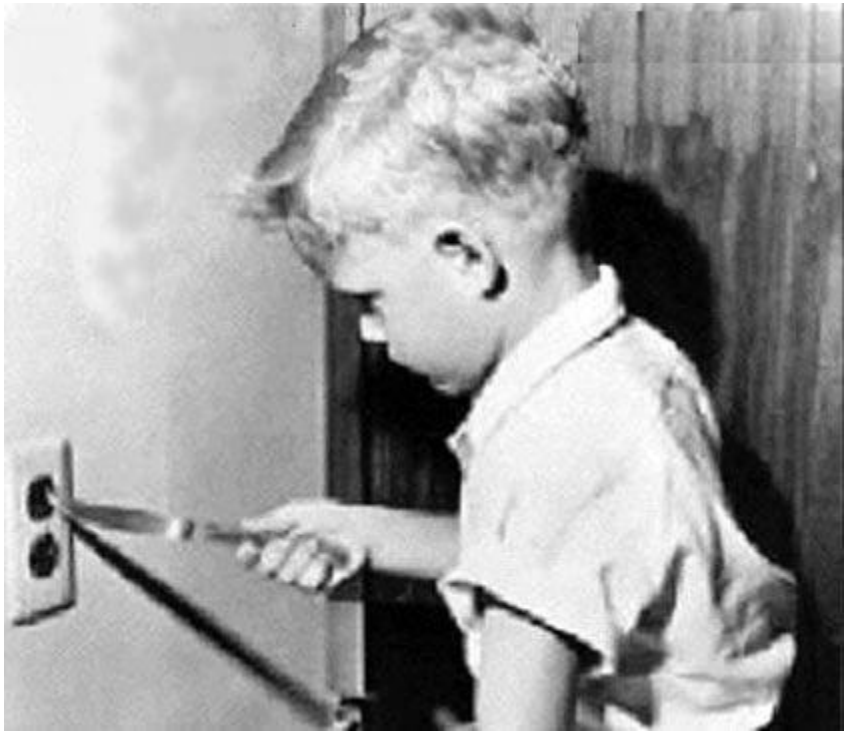
Only trying to get the software “correct” or to make it reliable will not make it safer under these conditions

Software Allows Unlimited System Complexity

- Complexity (**coupling**) means can no longer
  - Plan, understand, anticipate, and guard against all undesired system behavior
  - Exhaustively test to get out all design errors
- **Context** determines whether software is safe
  - Ariane 4 software was safe but when reused in Ariane 5, the spacecraft exploded
  - DAL, Rigor of Development, SIL will not ensure software is safe
  - Not possible to look at software alone and determine its "safety"

Safe or Unsafe?

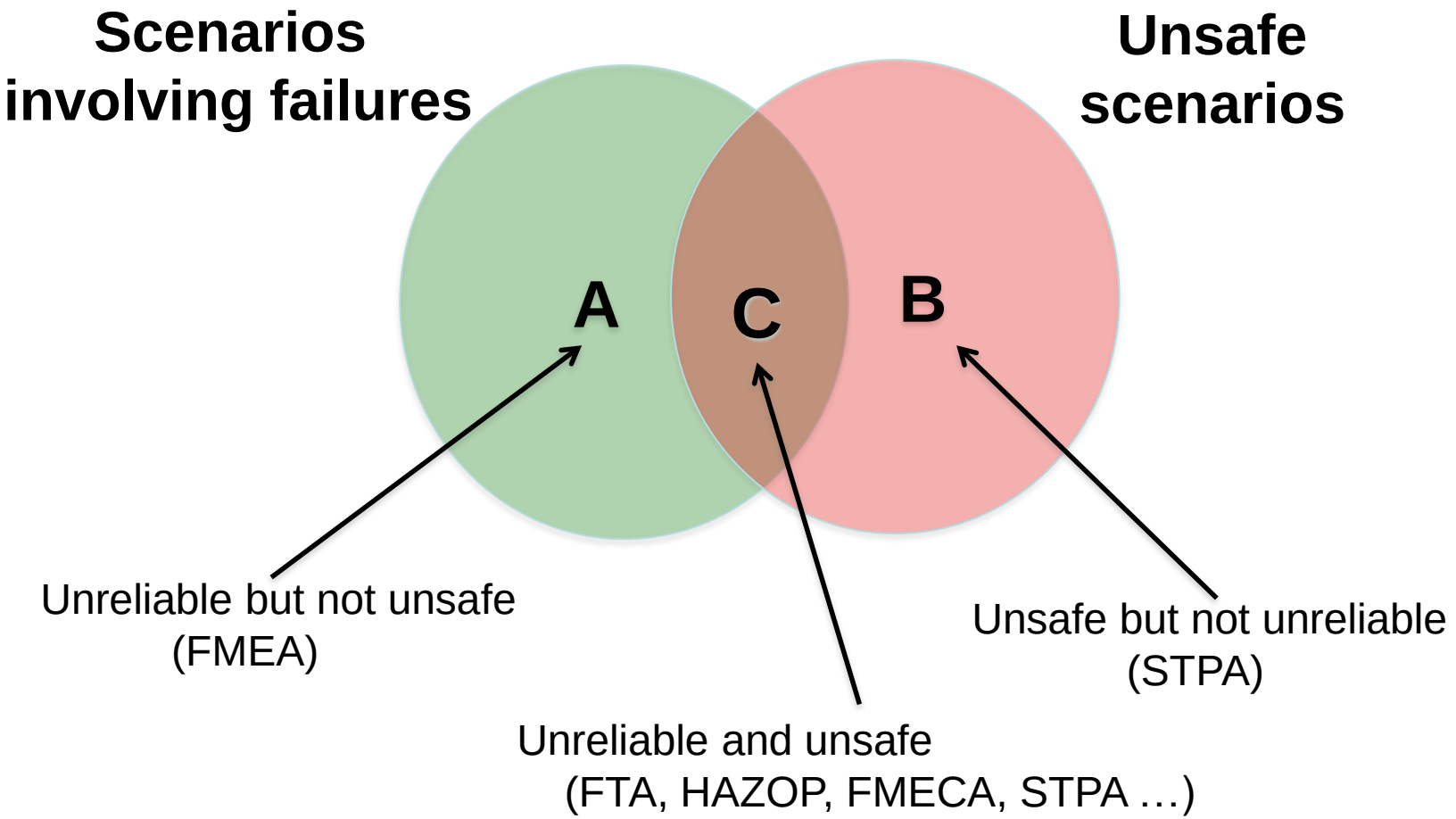
Safety Depends on Context



Washington State Ferry Problem

- Rental cars could not be driven off ferries when got to port
- Local rental car company installed a security device to prevent theft by disabling cars if car moved when engine stopped
- When ferry moved and cars not running, disabled them.

Confusing Safety and Reliability



Preventing Component or Functional Failures is Not Enough

Warsaw A320 Accident

- Software protects against activating thrust reversers when airborne
- Hydroplaning and other factors made the software not think the plane had landed
- Pilots could not activate the thrust reversers and ran off end of runway into a small hill.

Software changes the role of humans in systems

Typical assumption is that operator error is cause of most incidents and accidents

- So do something about operator involved (admonish, fire, retrain them)
- Or do something about operators in general
  - Marginalize them by putting in more automation
  - Rigidify their work by creating more rules and procedures

Another Accident Involving Thrust Reversers

- Tu-204, Moscow, 2012
- Red Wings Airlines Flight 9268
- The soft 1.12g touchdown made runway contact a little later than usual.
- With the crosswind, this meant weight-on-wheels switches did not activate and the thrust-reverse system would not deploy.

Another Accident Involving Thrust Reversers

- Pilots believe the thrust reversers are deploying like they always do. With the limited runway space, they quickly engage high engine power to stop quicker. Instead this accelerated the Tu-204 forwards, eventually colliding with a highway embankment.

Another Accident Involving Thrust Reversers

- Pilots believe the thrust reversers are deploying like they always do. With the limited runway space, they quickly engage high engine power to stop quicker. Instead this accelerates the Tu-204 forwards, eventually colliding with a highway embankment.

In complex systems, human and technical considerations cannot be isolated

A Systems View of Operator Error

- Operator error is a symptom, not a cause
- All behavior affected by context (system) in which occurs
  - Role of operators is changing in software-intensive systems as is the errors they make
  - Designing systems in which operator error inevitable and then blame accidents on operators rather than designers
- To do something about operator error, must look at system in which people work:
  - Design of equipment
  - Usefulness of procedures
  - Existence of goal conflicts and production pressures
- **Human error is a symptom of a system that needs to be redesigned**

Human factors concentrates on the “screen out”

Hardware/Software engineering concentrates on the “screen in”

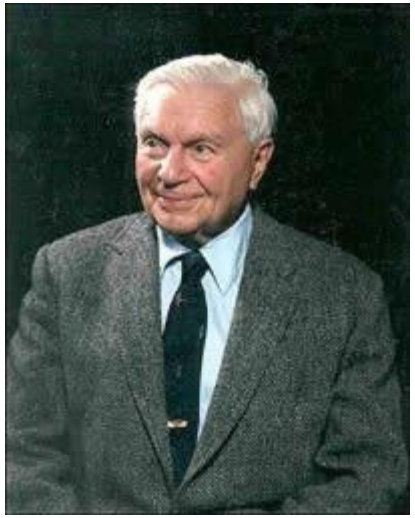
Not enough attention on integrated system as a whole

(e.g, mode confusion, situation awareness errors, inconsistent behavior, etc.

Jerome Lederer (1968)

"Systems safety covers the total spectrum of risk management. It goes *beyond the hardware* and associated procedures of systems safety engineering. It involves:

- Attitudes and motivation of designers and production people,
- Employee/management rapport,
- The relation of industrial associations among themselves and with government,
- Human factors in supervision and quality control
- The interest and attitudes of top management



- The effects of the legal system on accident investigations and exchange of information
- The certification of critical workers
- Political considerations
- Resources
- Public sentiment

And many other non-technical but vital influences on the attainment of an acceptable level of risk control. These non-technical aspects of system safety cannot be ignored.”

We Need Something New

- New levels of complexity do not fit into a reliability-oriented world.
- Two approaches being taken now:

Pretend there is no problem

Shoehorn new technology and new levels of complexity into old methods

Summary of the Problem:

- We need models and tools that include:
  - Hardware and hardware failures
  - Software (particularly requirements)
  - Human factors
  - Interactions among system components
  - System design errors
  - Management, regulation, policy
  - Environmental factors

  and the “unknown unknowns”

What is STAMP and how does it differ from what people do now?

The Problem is Complexity

Ways to Cope with Complexity

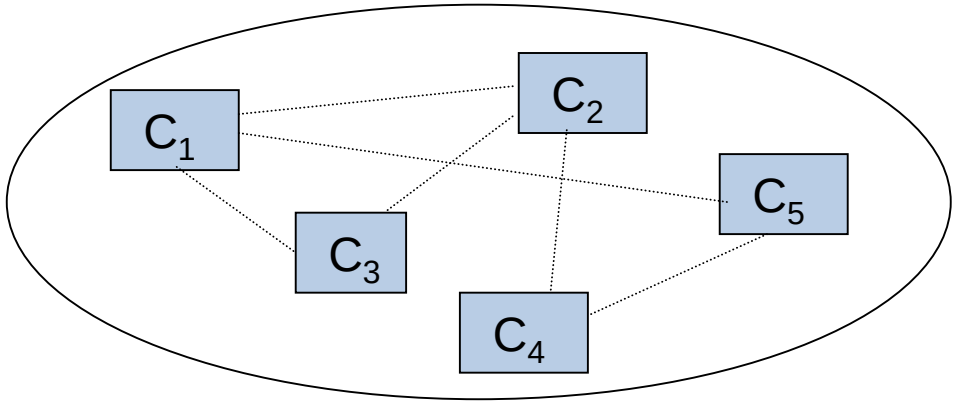
- Analytic Reduction
- Statistics
- Systems Theory

Traditional Approach to Coping with Complexity

Analytic Reduction ("Divide and Conquer")

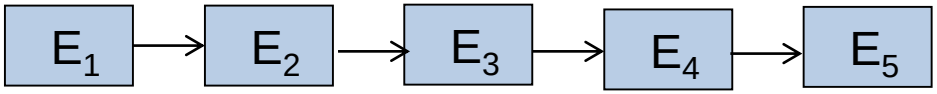
1. Divide system into separate parts

Physical/Functional: Separate into distinct components



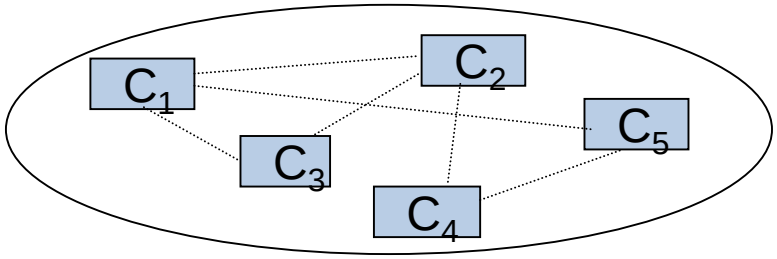
Components interact
In direct ways

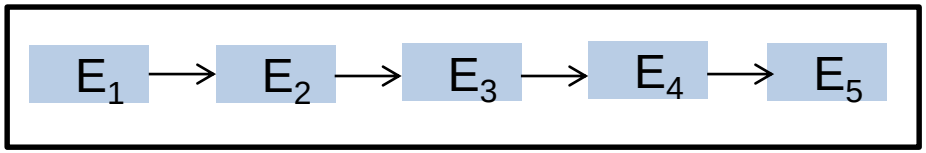
Behavior: Separate into events over time



Each event is the direct
result of the preceding event

Analytic Reduction (2)





2. Analyze/examine pieces separately and combine results

- Assumes such separation does not distort phenomenon
  - ✓ Each component or subsystem operates independently
  - ✓ Components act the same when examined singly as when playing their part in the whole
  - ✓ Components/events not subject to feedback loops and non-linear interactions
  - ✓ Interactions can be examined pairwise

Bottom Line

- These assumptions are no longer true in our
  - Tightly coupled
  - Software intensive
  - Highly automated
  - Connected

  engineered systems
- Need a new theoretical basis
  - *System theory* can provide it

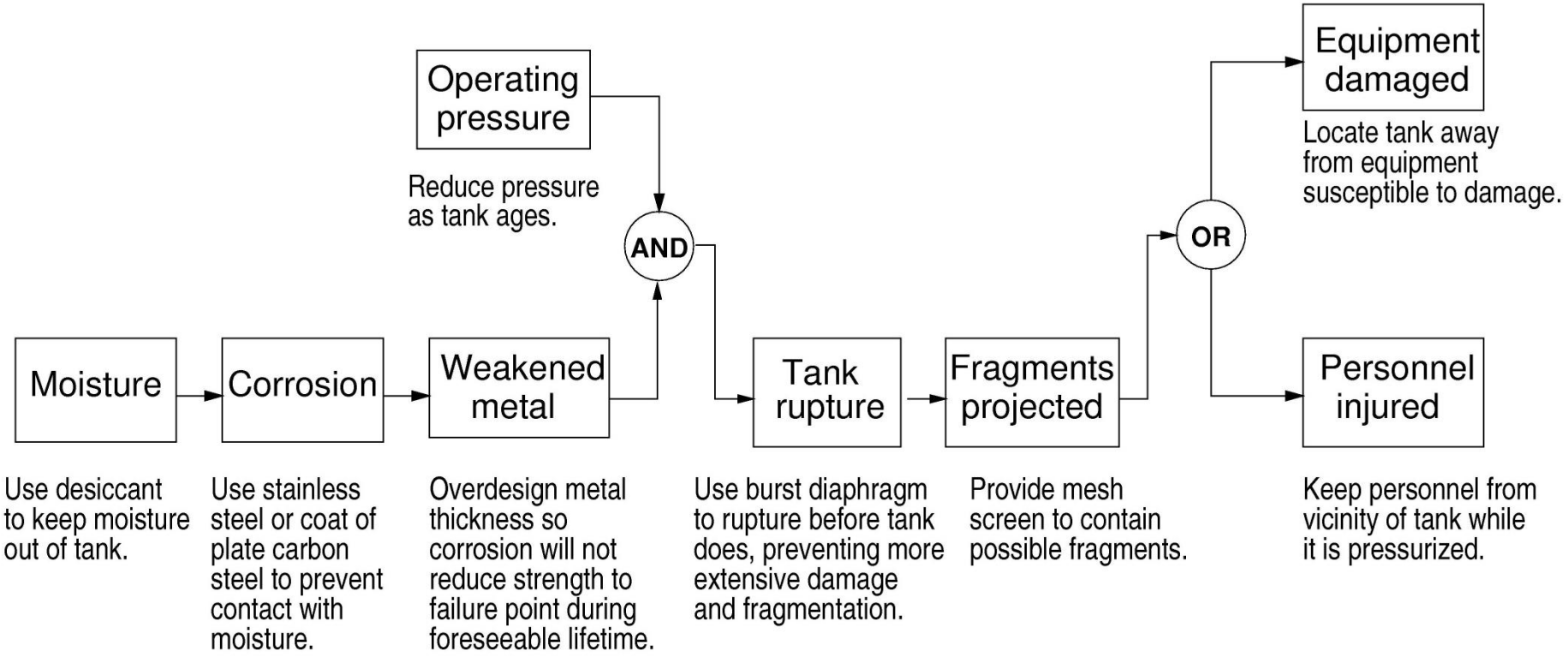
Traditional Approach to Safety

- Reductionist
  - Divide system into components
  - Assume accidents are caused by component failure
  - Identify chains of directly related physical or logical (functional) component failures that can lead to a loss
  - Evaluate reliability of components separately and later combine analysis results into a system reliability value

Note: Assume randomness in the failure events so can derive probabilities for a loss

- **Software and humans do not satisfy this assumption**

Chain-of-events example



Accidents as Chains of Failure Events

- Forms the basis for most safety engineering and reliability engineering analysis:

  FTA, PRA, FMEA/FMECA, Event Trees, FHA, etc.

  and design (concentrate on dealing with component failure):

  Redundancy and barriers (to prevent failure propagation)

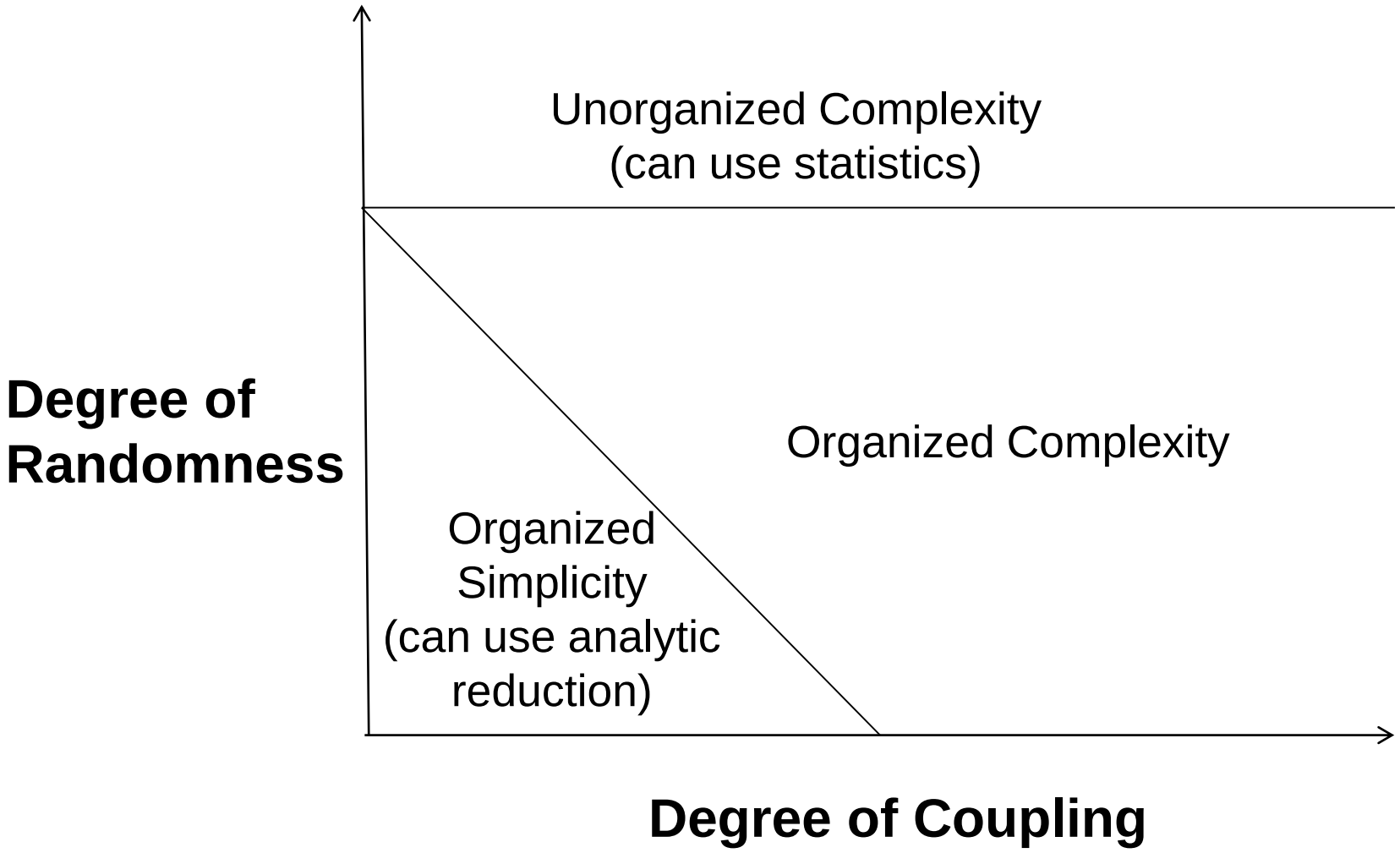
  High component integrity and overdesign

  Fail-safe design

  (humans) Operational procedures, checklists, training, ….

Standard Approach does not Handle

- Component interaction accidents
- Systemic factors (affecting all components and barriers)
- Software and software requirements errors
- Human behavior (in a non-superficial way)
- System design errors
- Indirect or non-linear interactions and complexity
- Migration of systems toward greater risk over time (e.g., in search for greater efficiency and productivity)


Unorganized Complexity
(can use statistics)
Degree of
Randomness
Organized Complexity
Organized
Simplicity
(can use analytic
reduction)
Degree of Coupling

Systems Theory

- Developed for systems that are
  - Too complex for complete analysis
    - Separation into (interacting) subsystems distorts the results
    - The most important properties are emergent
  - Too organized for statistics
    - Too much underlying structure that distorts the statistics
    - New technology and designs have no historical information
- First used on ICBM systems of 1950s/1960s
- Basis for System Engineering and System Safety

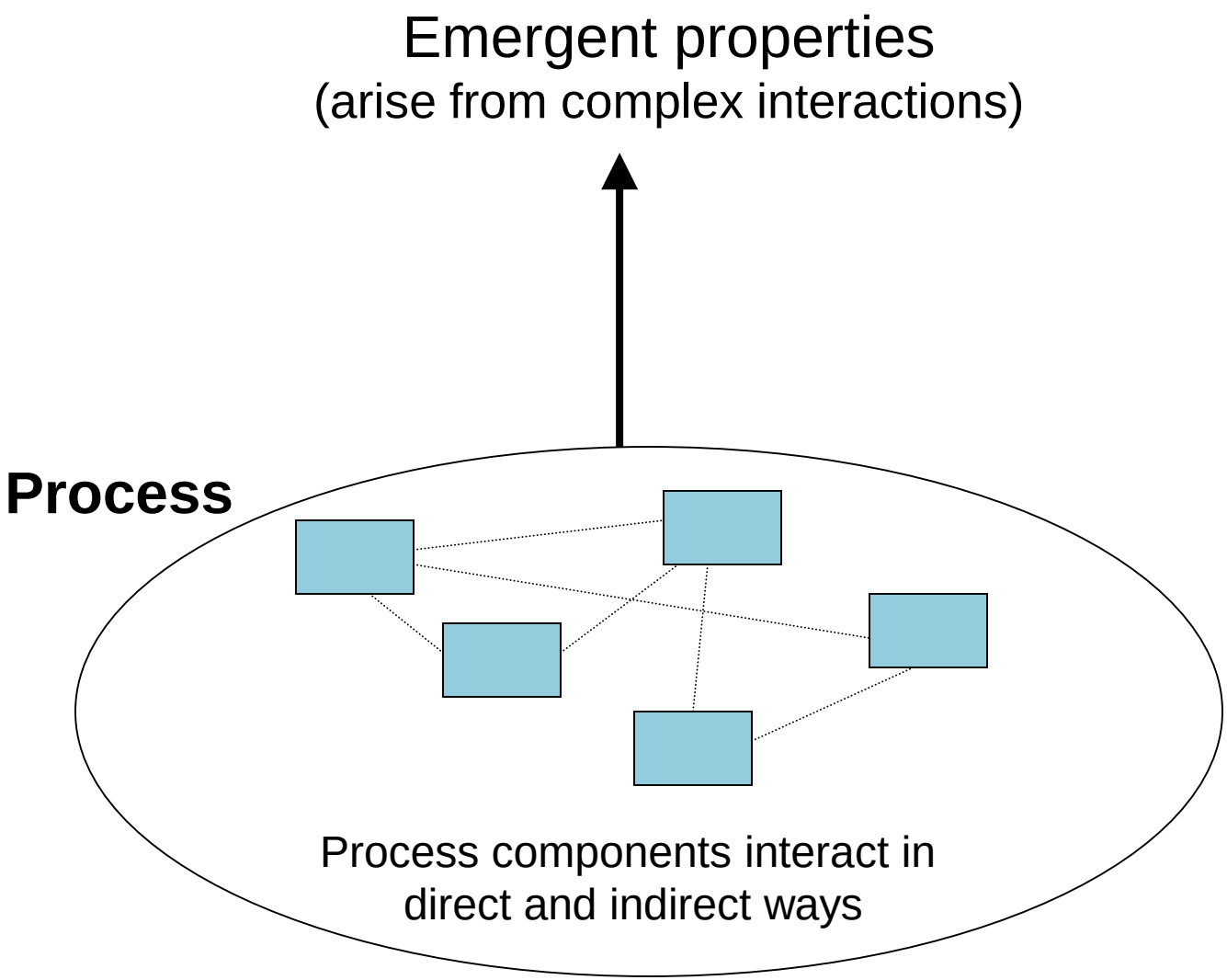
Systems Theory (2)

- Focuses on systems taken as a whole, not on parts taken separately
- Emergent properties
  - Some properties can only be treated adequately in their entirety, taking into account all social and technical aspects

    "The whole is greater than the sum of the parts"

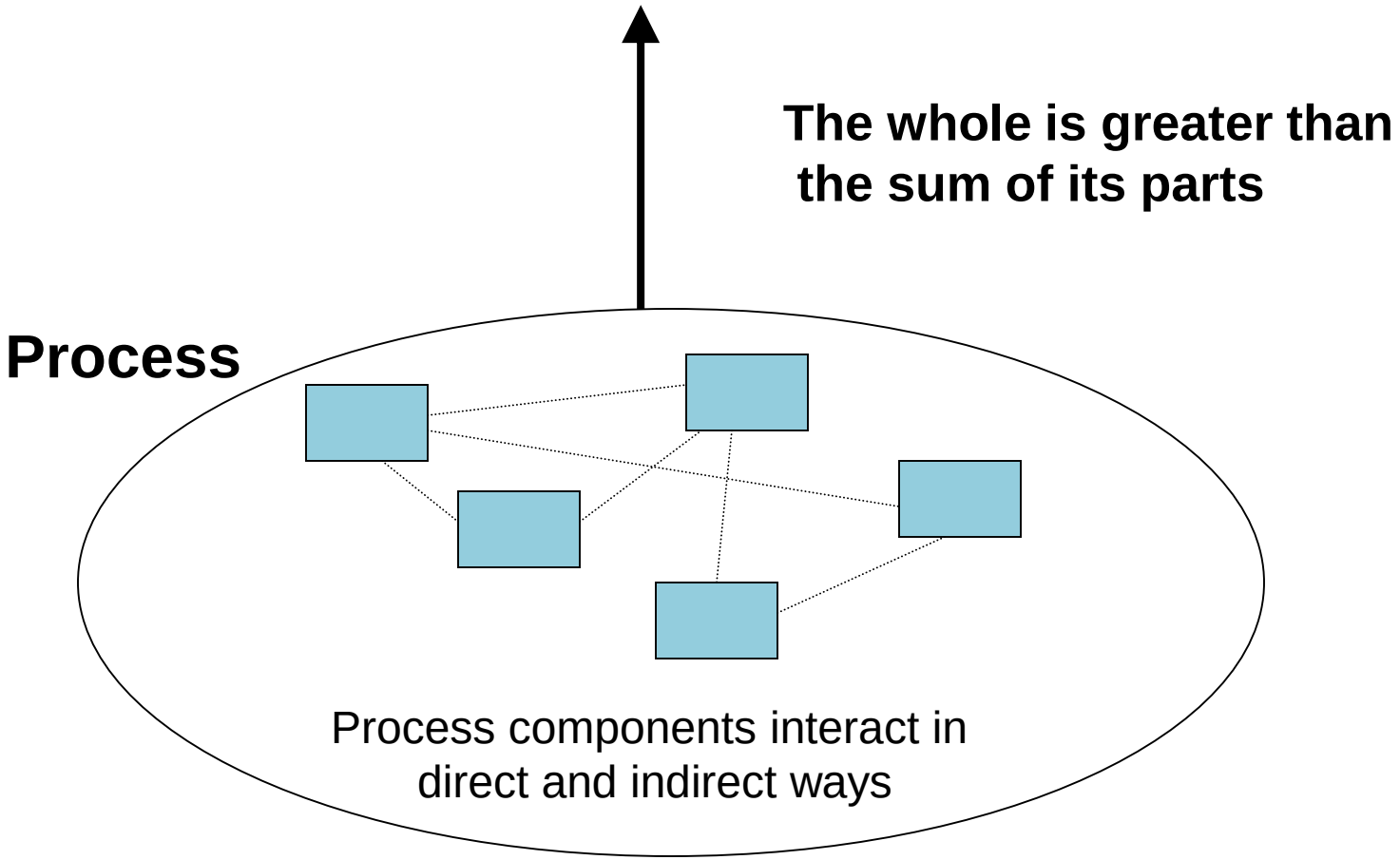
  - These properties arise from relationships among the parts of the system

    How they interact and fit together

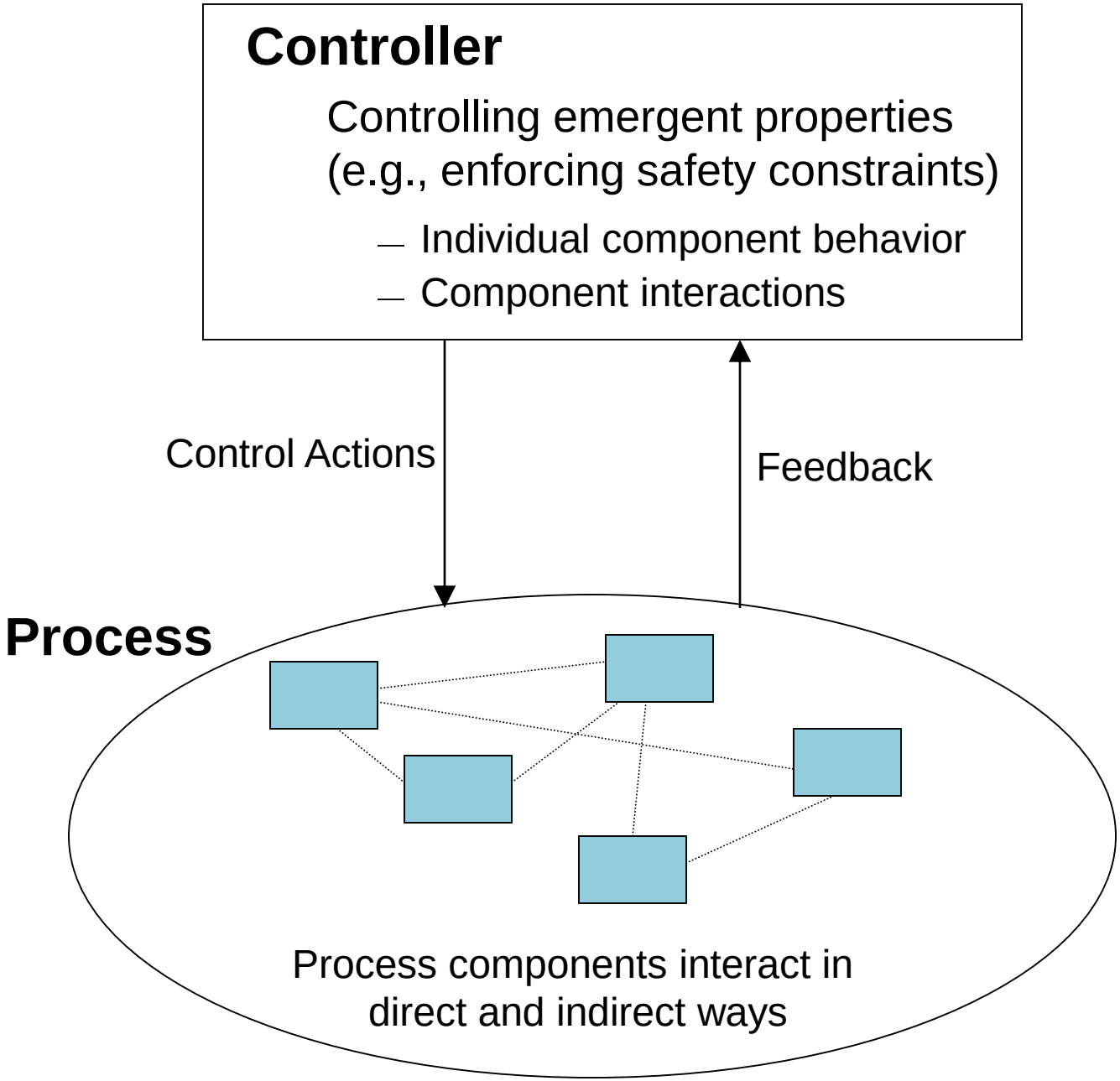


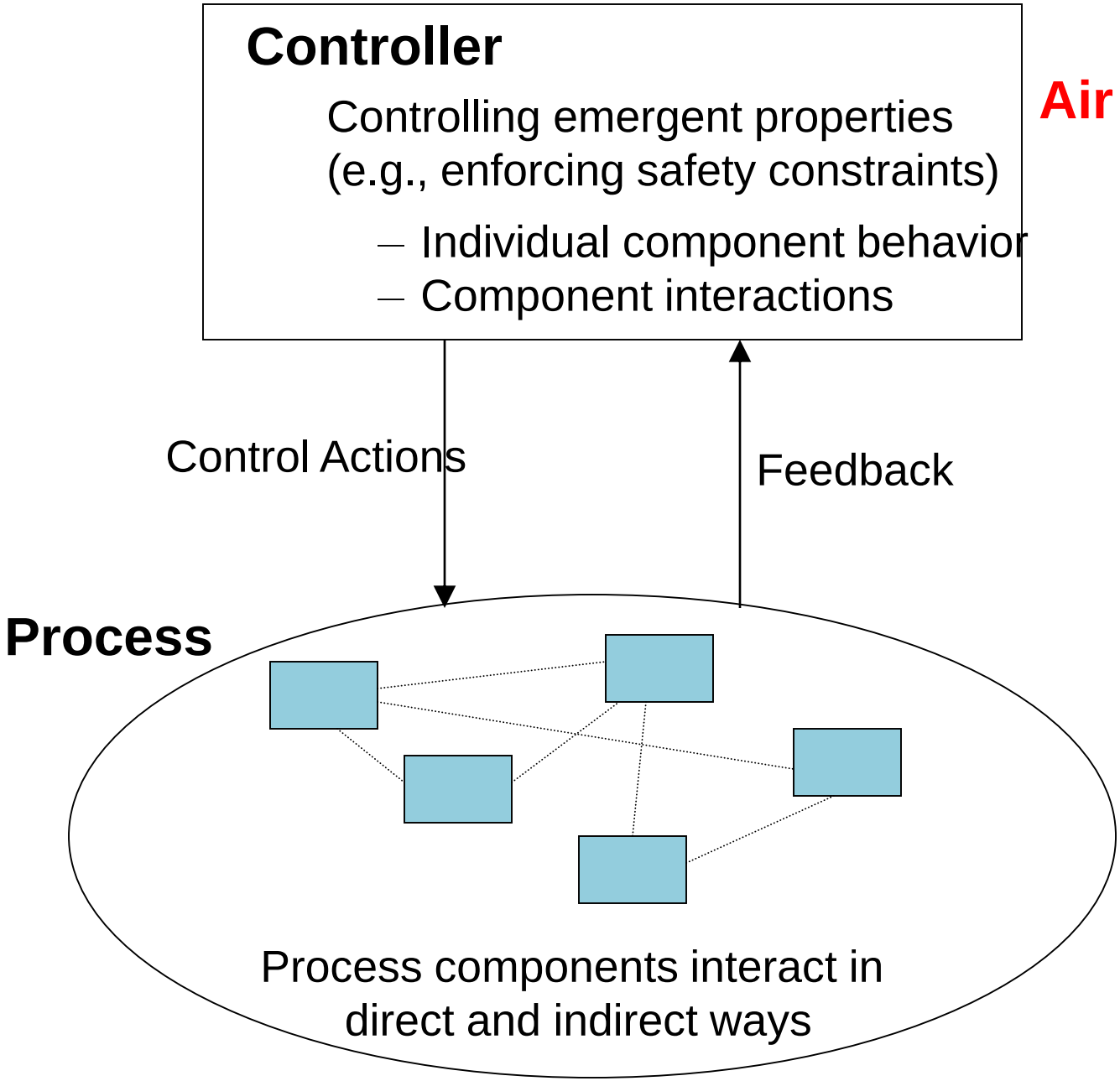
Safety and security are emergent properties

Emergent properties
(arise from complex interactions)



Safety and security are emergent properties


Controller
Controlling emergent properties
(e.g., enforcing safety constraints)
– Individual component behavior
– Component interactions
Control Actions
Feedback
Process
Process components interact in
direct and indirect ways



Air Traffic Control:
Safety
Throughput

Controls/Controllers Enforce Safety Constraints

- Power must never be on when access door open
- Two aircraft/automobiles must not violate minimum separation
- Aircraft must maintain sufficient lift to remain airborne
- Integrity of hull must be maintained on a submarine
- Toxic chemicals/radiation must not be released from plant
- Workers must not be exposed to workplace hazards
- Public health system must prevent exposure of public to contaminated water and food products
- Pressure in a offshore well must be controlled

Controls/Controllers Enforce Safety Constraints (2)

- Runway incursions and operations on wrong runways or taxiways must be prevented
- Bomb must not detonate without positive action by authorized person
- Submarine must always be able to blow the ballast tanks and return to surface
- Truck drivers must not drive when sleep deprived
- Fire must not be initiated on a friendly target

These are the High-Level Functional Safety Requirements to Address During Design

A Broad View of "Control"

Component failures and unsafe interactions may be "controlled" through design

(e.g., redundancy, interlocks, fail-safe design)

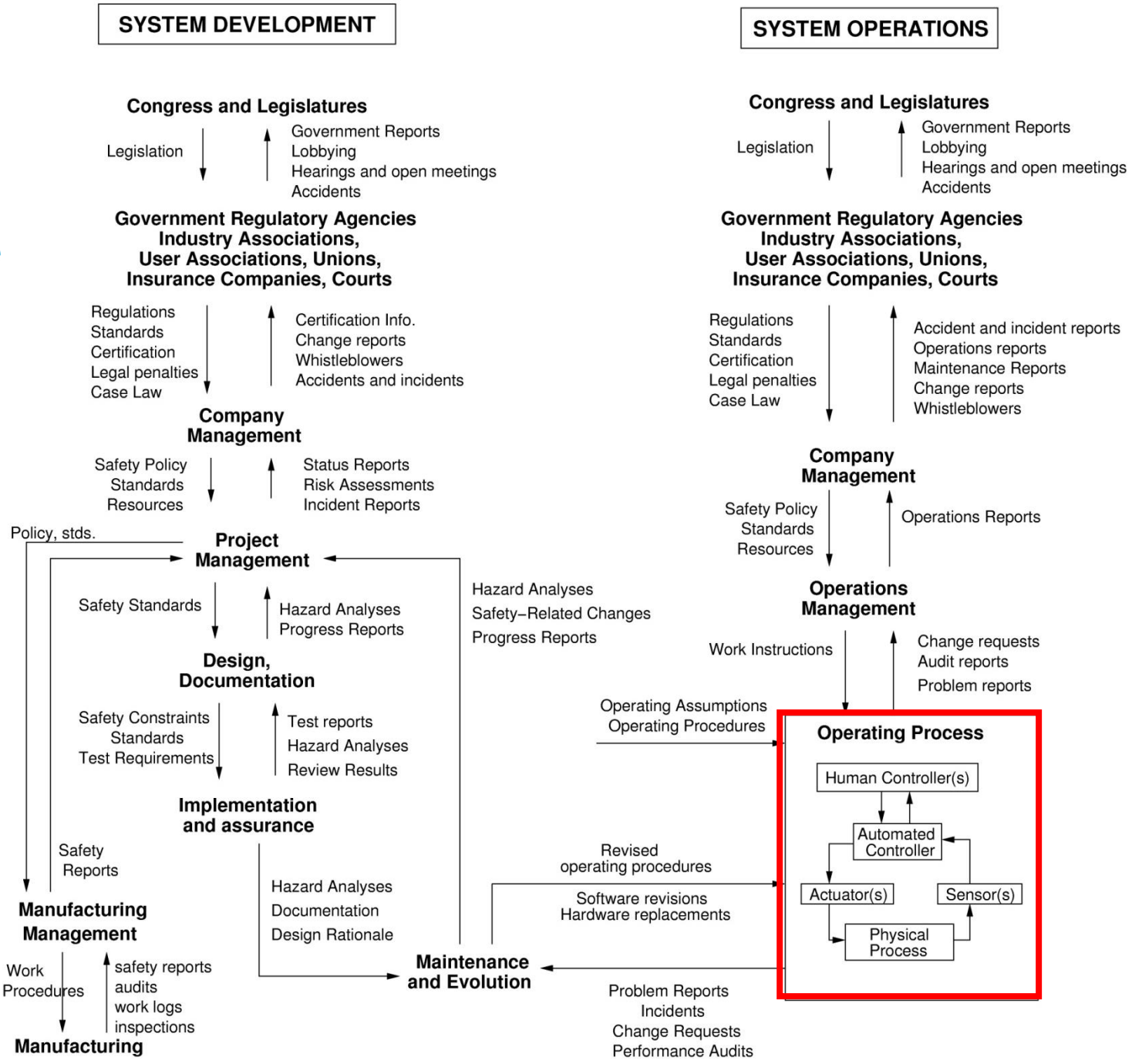
or through process

- Manufacturing processes and procedures
- Maintenance processes
- Operations

or through social controls

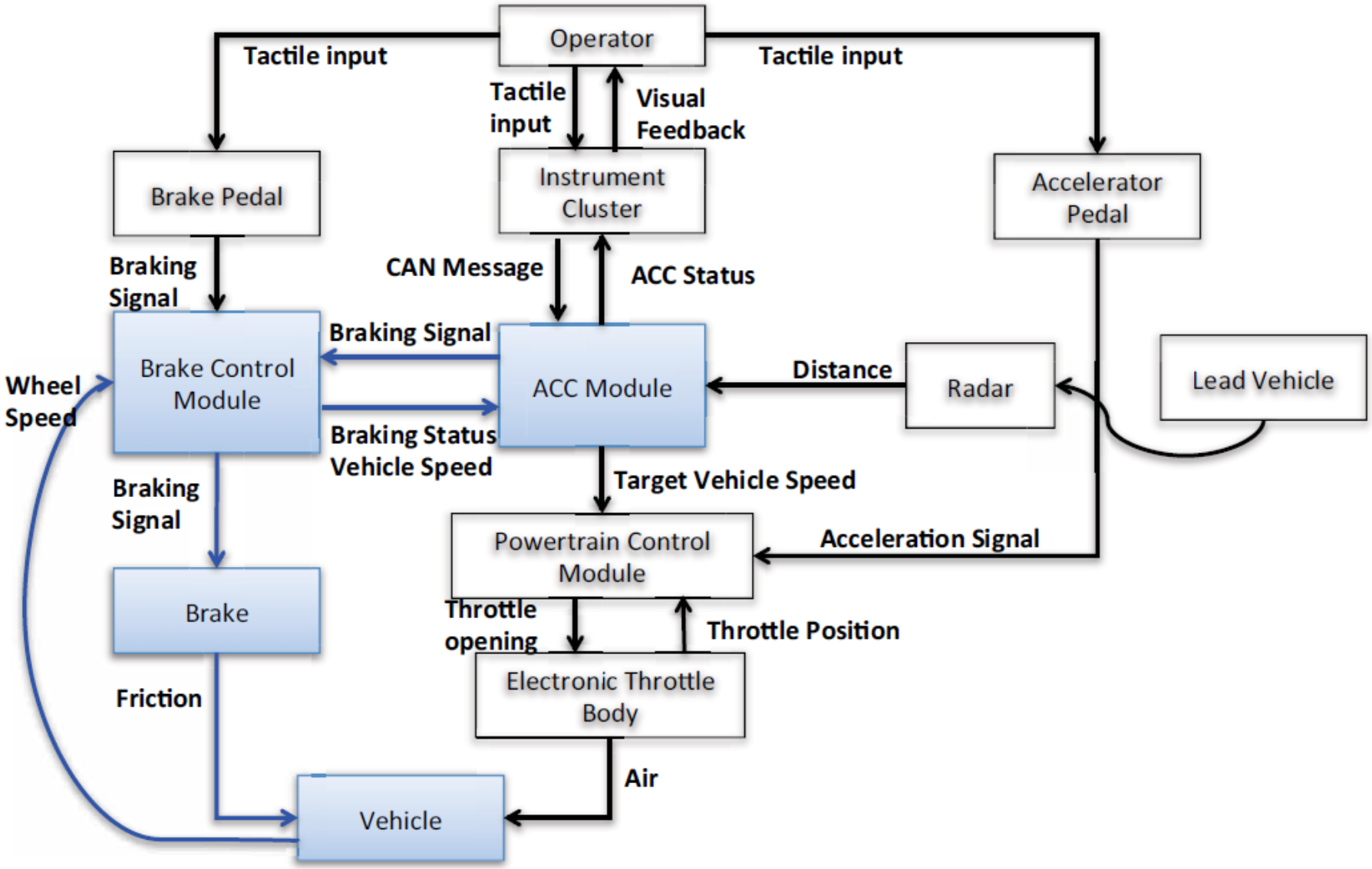
- Governmental or regulatory
- Culture
- Insurance
- Law and the courts
- Individual self-interest (incentive structure)

Example Safety Control Structure (SMS)



(Qi Hommes)

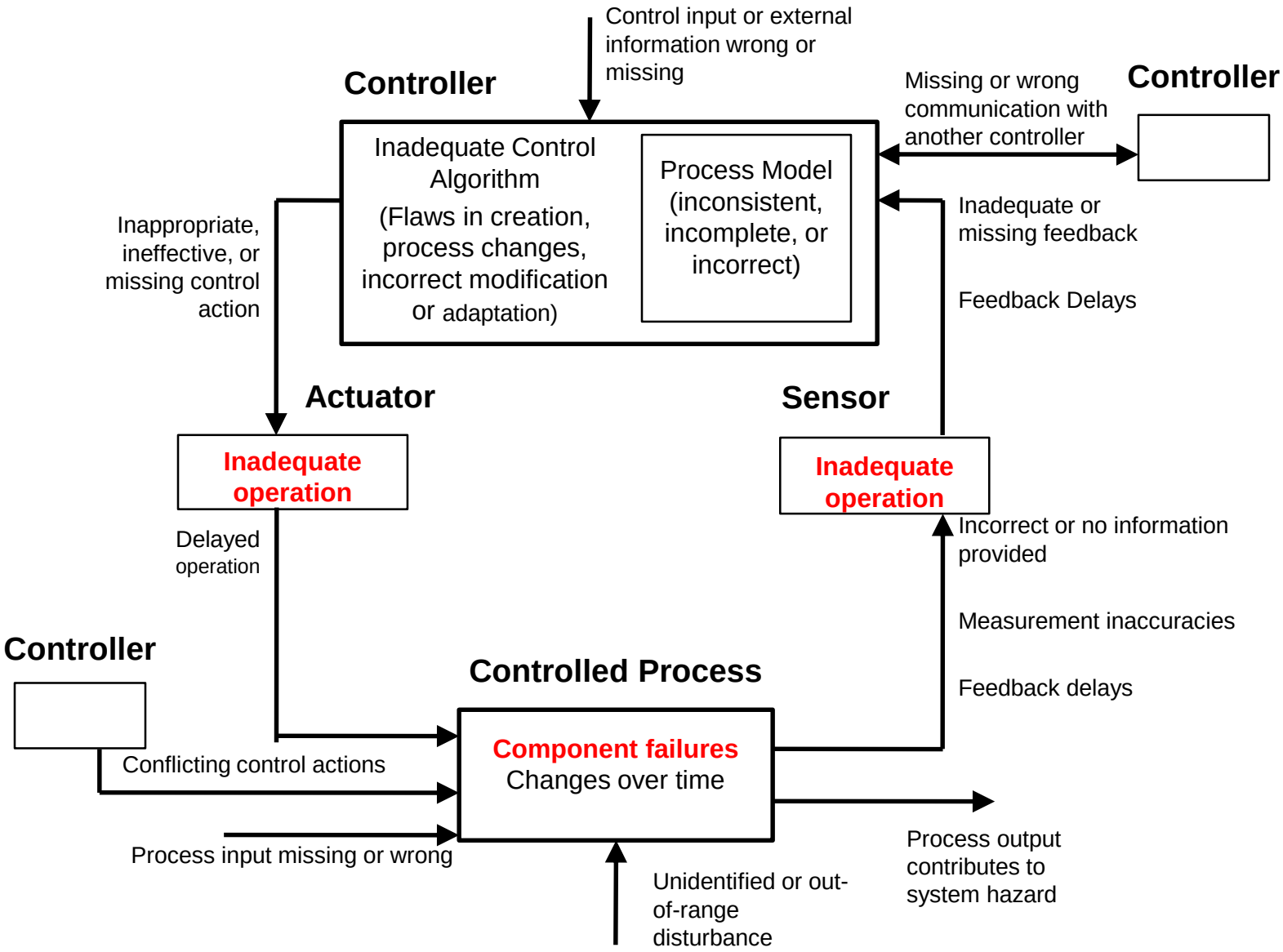
Example: ACC – BCM Control Loop



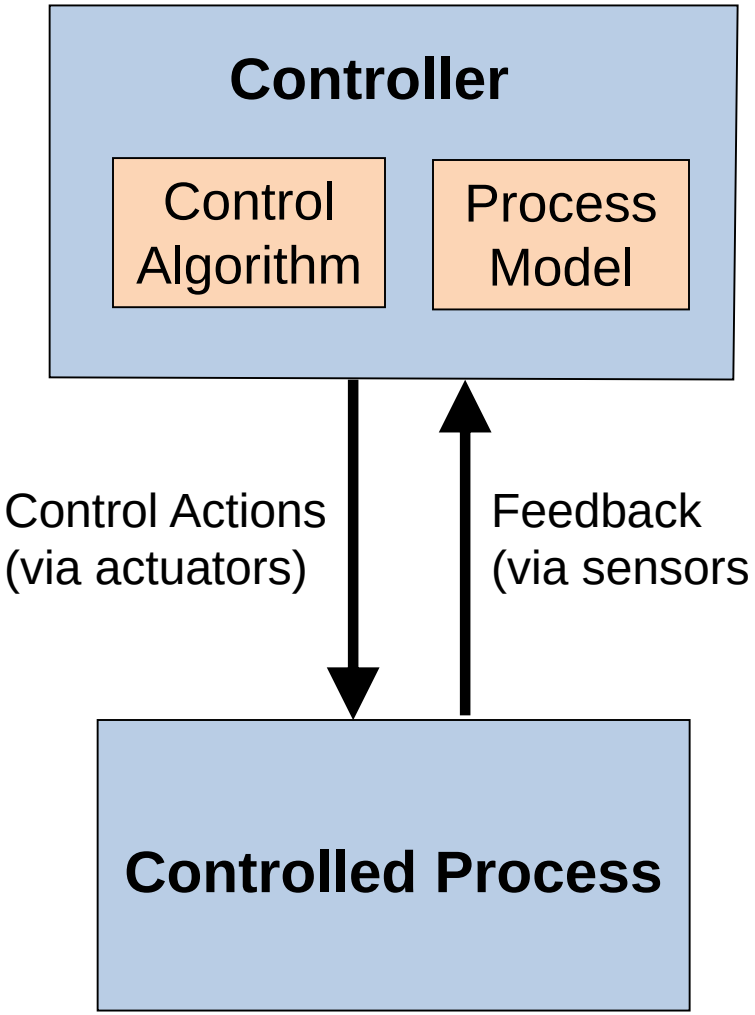
Safety as a Control Problem

- **Goal: Design an effective control structure that eliminates or reduces adverse events**.
  - Need clear definition of expectations, responsibilities, authority, and accountability at all levels of safety control structure
  - Need appropriate feedback
  - Entire control structure must together enforce the system safety property (constraints)
    - Physical design (inherent safety)
    - Operations
    - Management
    - Social interactions and culture

Identifying Causal Scenarios for Unsafe Control

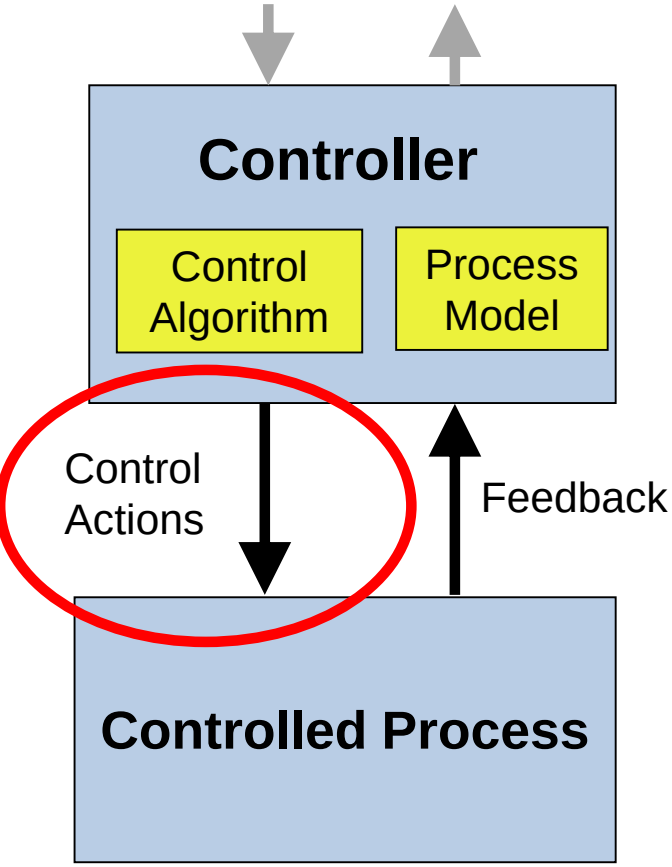


Role of Process Models in Control



- Controllers use a **process model** to determine control actions
- Software/human related accidents often occur when the process model is incorrect
- Captures software errors, human errors, flawed requirements …

Unsafe Control Actions

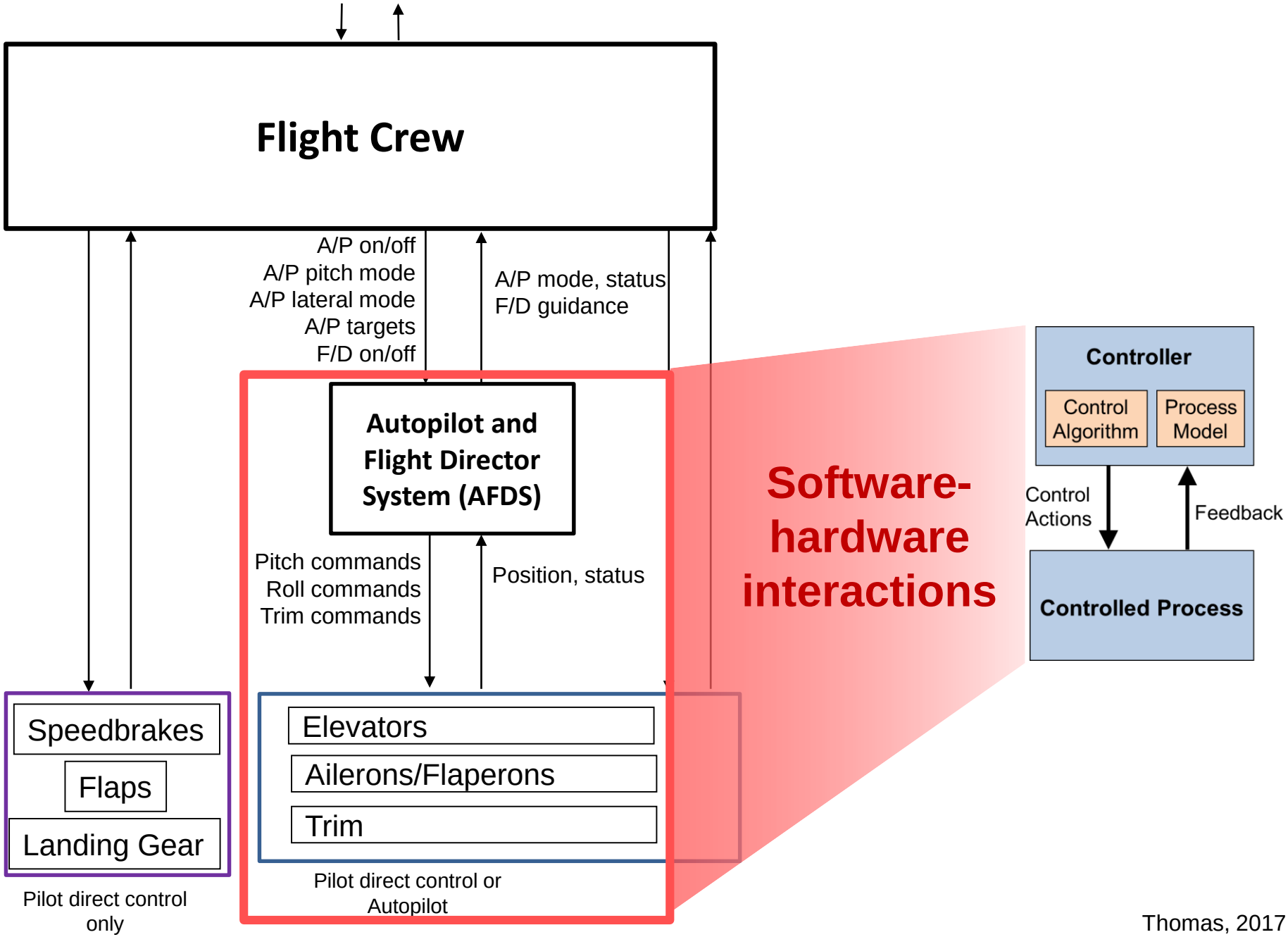


Four types of unsafe control actions

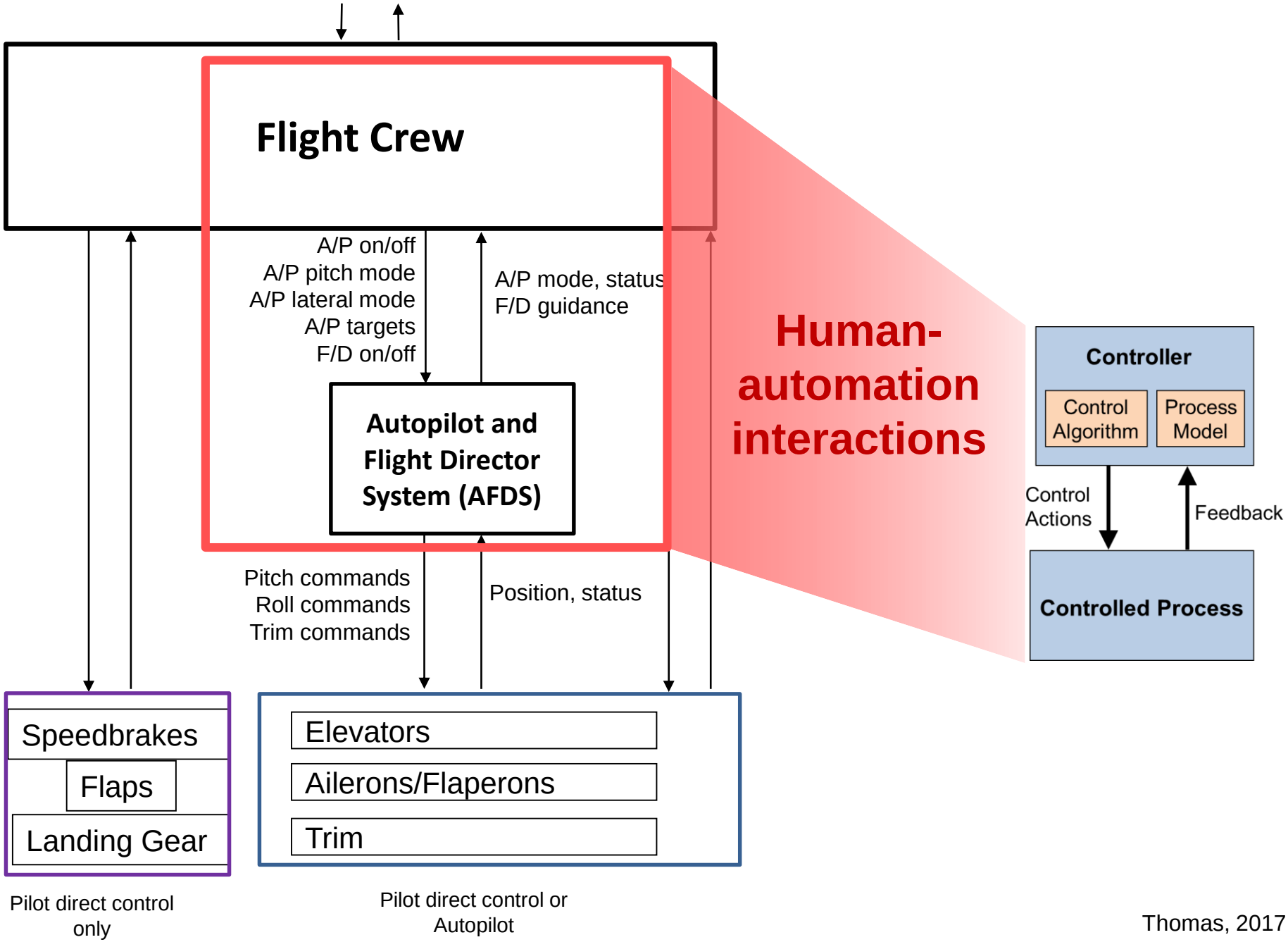
1) Control commands required for safety are not given
2) Unsafe commands are given
3) Potentially safe commands but given too early, too late
4) Control action stops too soon or applied too long (continuous control)

Analysis:

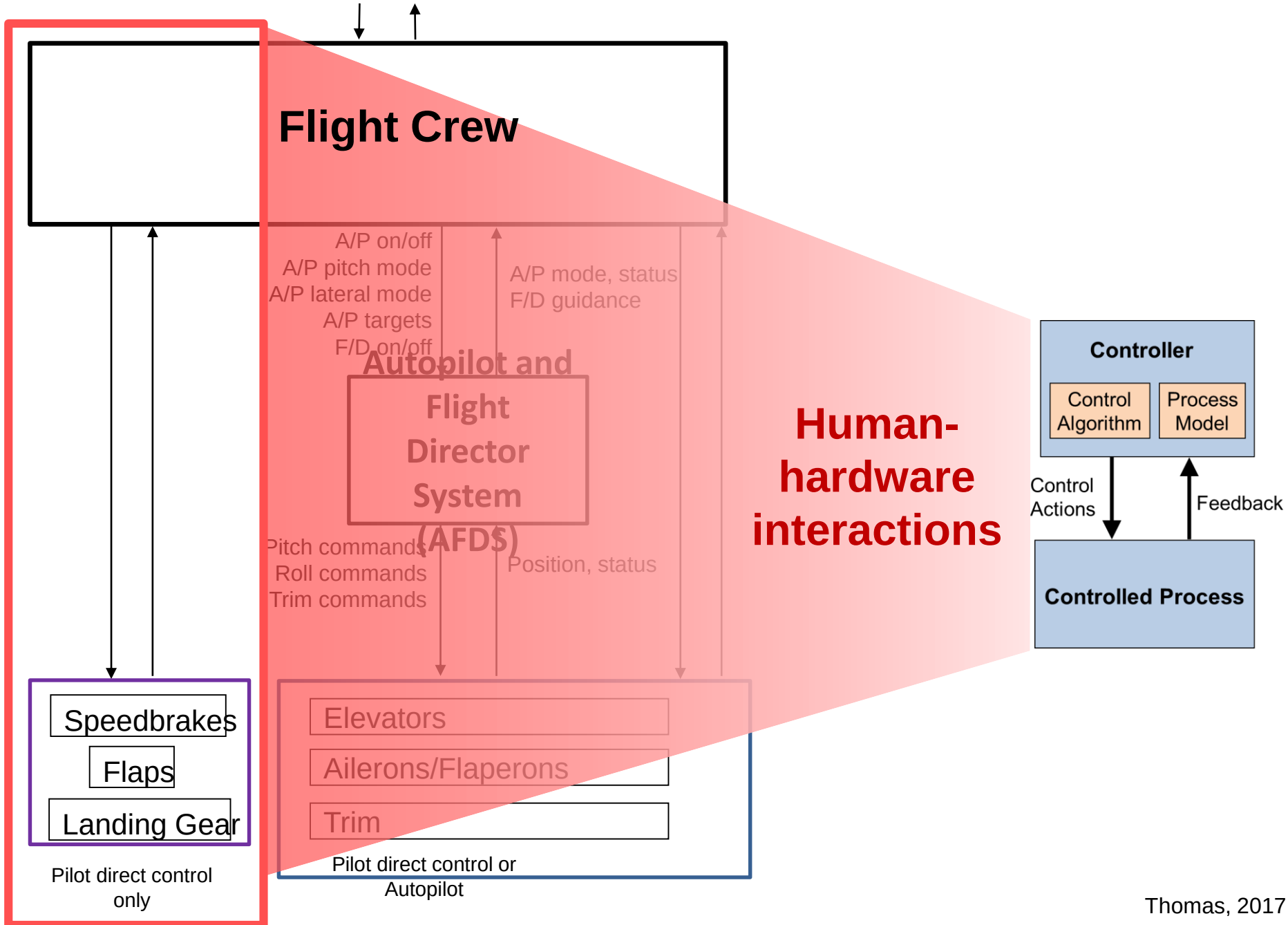
1. Identify potential unsafe control actions
2. Identify why they might be given
3. If safe ones provided, then why not followed?


Flight Crew
A/P on/off
A/P pitch mode
A/P lateral mode
A/P targets
F/D on/off
A/P mode, status
F/D guidance
Autopilot and Flight Director System (AFDS)
Pitch commands
Roll commands
Trim commands
Position, status
Speedbrakes
Flaps
Landing Gear
Elevators
Ailerons/Flaperons
Trim
Pilot direct control only
Pilot direct control or Autopilot
Software-hardware interactions
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process

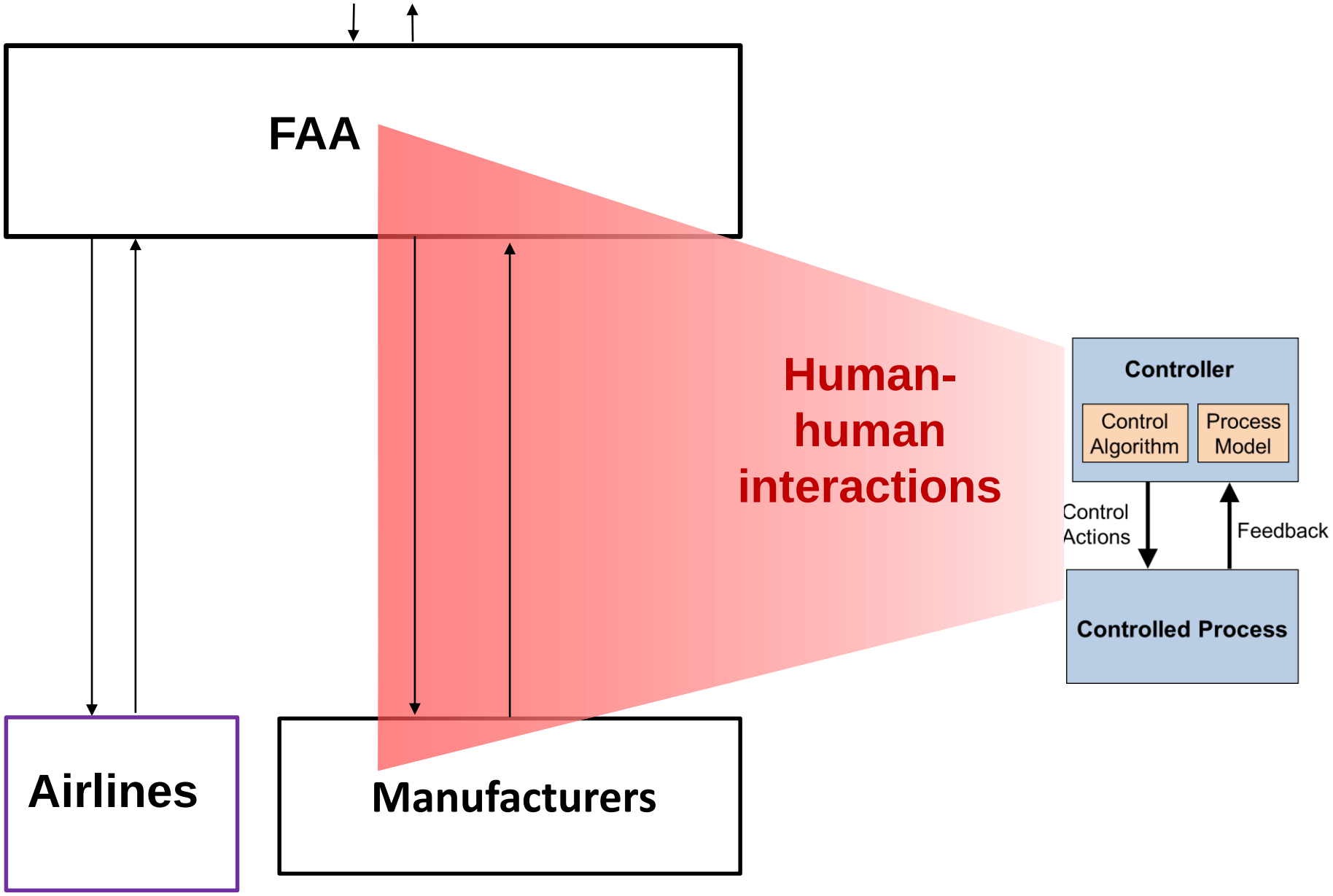
Thomas, 2017


Flight Crew
A/P on/off
A/P pitch mode
A/P lateral mode
A/P targets
F/D on/off
A/P mode, status
F/D guidance
Autopilot and Flight Director System (AFDS)
Pitch commands
Roll commands
Trim commands
Position, status
Speedbrakes
Flaps
Landing Gear
Elevators
Ailerons/Flaperons
Trim
Pilot direct control only
Pilot direct control or Autopilot
Human-automation interactions
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process

Thomas, 2017


Flight Crew
A/P on/off
A/P pitch mode
A/P lateral mode
A/P targets
F/D on/off
A/P mode, status
F/D guidance
Autopilot and Flight Director System (AFDS)
Pitch commands
Roll commands
Trim commands
Position, status
Human-hardware interactions
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process
Speedbrakes
Flaps
Landing Gear
Pilot direct control only
Elevators
Ailerons/Flaperons
Trim
Pilot direct control or Autopilot

Thomas, 2017


FAA
Human-
human
interactions
Controller
Control
Algorithm
Process
Model
Control
Actions
Feedback
Controlled Process
Airlines
Manufacturers

Thomas, 2017

STAMP (System-Theoretic Accident Model and Processes)

- Defines safety/security as a control problem (vs. failure problem)
- Applies to very complex systems
- Includes software, humans, operations, management, culture
- Based on general system theory
- Expands the traditional model of the accident causation (cause of losses)
  - Not just a chain of directly related failure events
  - Losses are complex processes

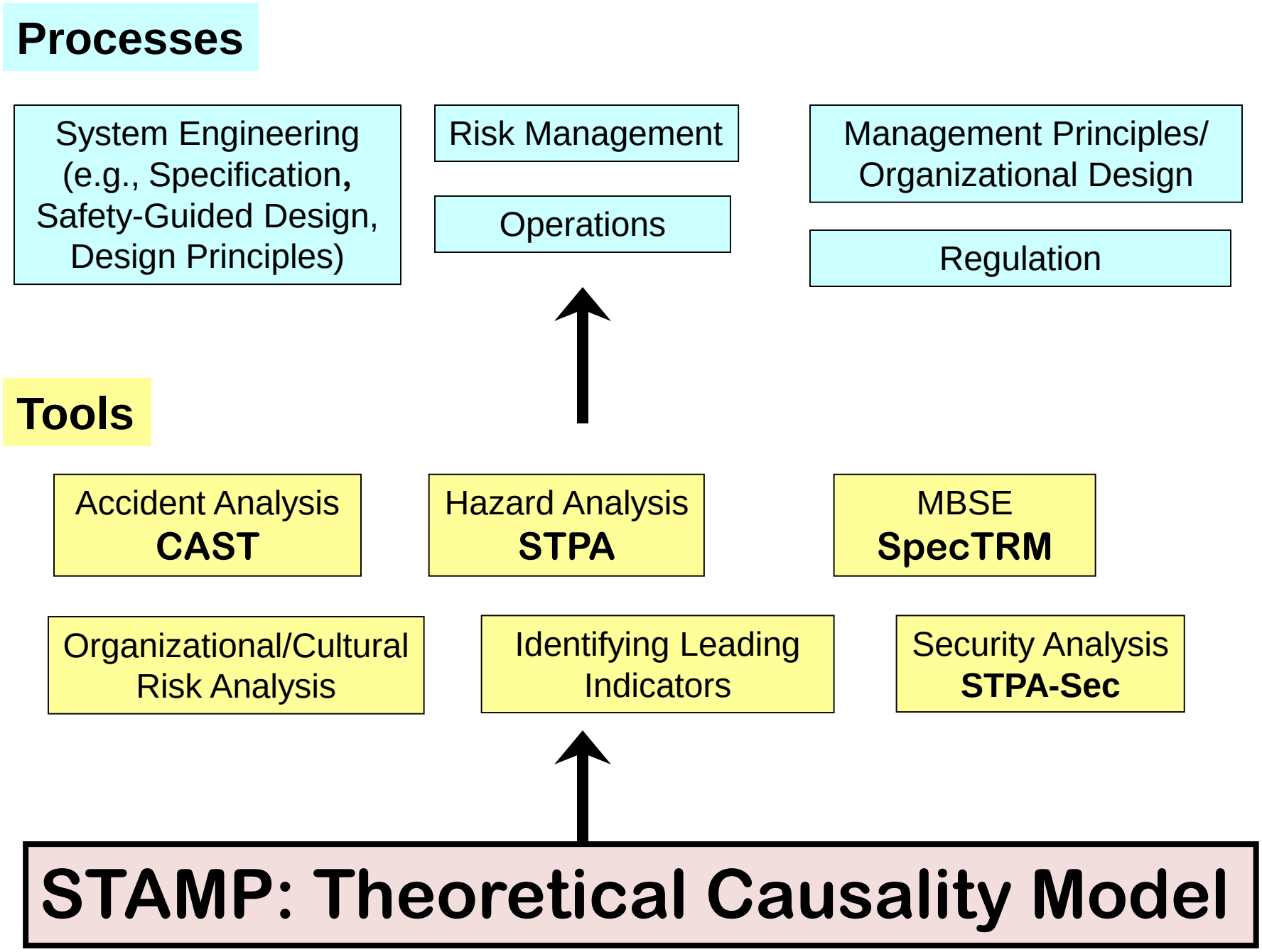
Safety as a Dynamic Control Problem (STAMP)

- Hazards result from lack of enforcement of safety constraints in system design and operations
- Goal is to control the behavior of the components and systems as a whole to ensure safety constraints are enforced in the operating system
- A change in emphasis:

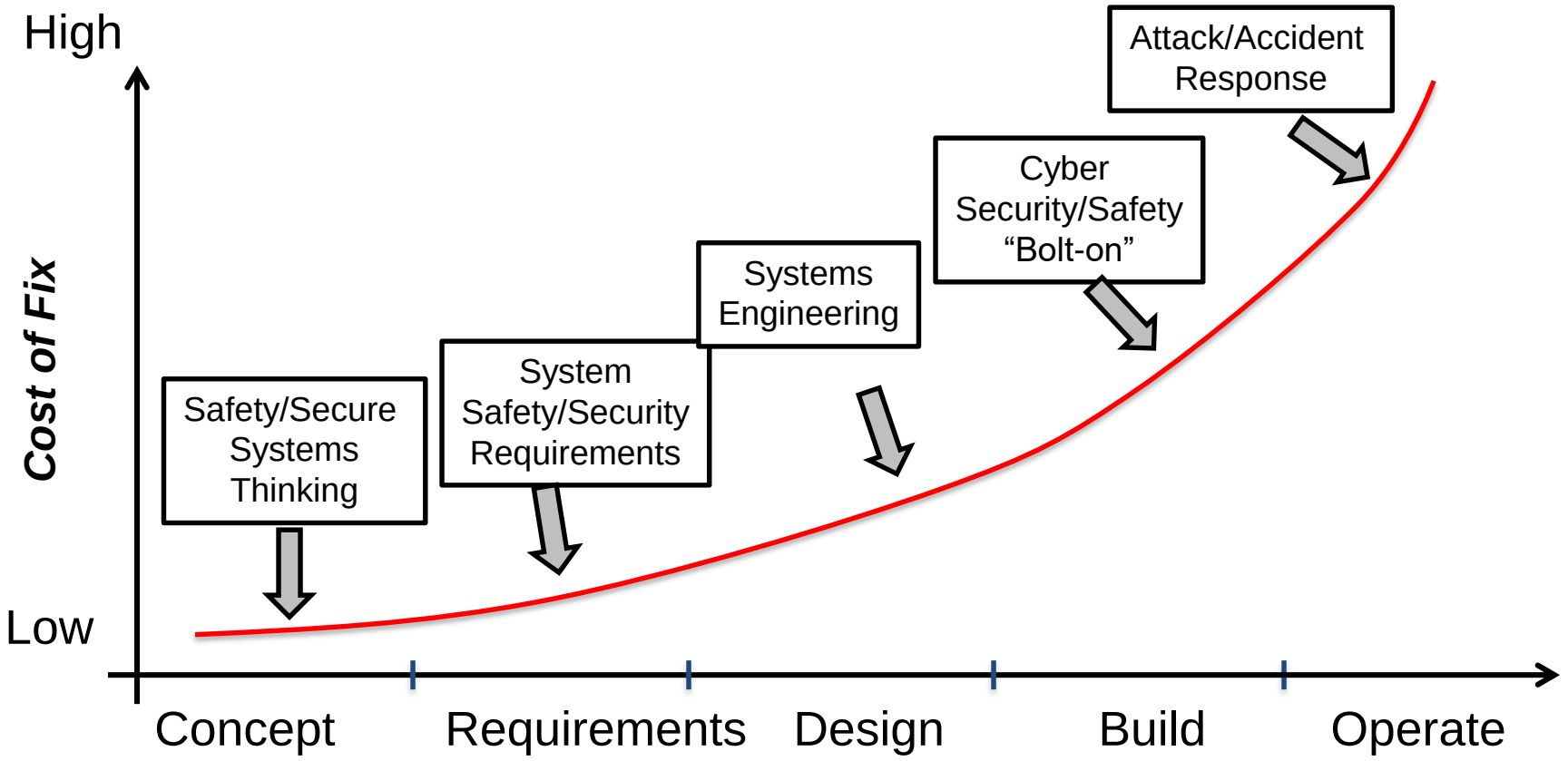
"enforce safety/security constraints on system behavior"

(note that enforcing constraints might require preventing failures or handling them but includes more than that)

What kinds of tools are available?


Processes
System Engineering (e.g., Specification, Safety-Guided Design, Design Principles)
Risk Management
Operations
Management Principles/ Organizational Design
Regulation
Tools
Accident Analysis CAST
Hazard Analysis STPA
MBSE SpecTRM
Organizational/Cultural Risk Analysis
Identifying Leading Indicators
Security Analysis STPA-Sec
STAMP: Theoretical Causality Model

Build safety and security into system from beginning



Integrated Approach to Safety and Security (Col. Bill Young)

- Safety: prevent losses due to unintentional actions by benevolent actors
- Security: prevent losses due to intentional actions by malevolent actors
- Key difference is intent
- Common goal: loss prevention
  - Ensure that critical functions and services provided by networks and services are maintained
  - New paradigm for safety will work for security too
    - May have to add new causes, but rest of process is the same
  - A top-down, system engineering approach to designing safety and security into systems

Integrated Approach to Safety and Security

- Both concerned with losses (intentional or unintentional)
- Starts with defining unacceptable losses
  - "What": essential services to be secured
  - "What" used later to reason thoroughly about "how" best to guard against threats
  - Analysis moves from general to specific
    - Less likely to miss things
    - Easier to review

Example: Stuxnet

- Loss: Damage to reactor (in this case centrifuges)
- Hazard/Vulnerability: Centrifuges are damaged by spinning too fast
- Constraint to be Enforced: Centrifuges must never spin above maximum speed
- Hazardous control action: Issuing *increase speed* command when already spinning at maximum speed
- One potential causal scenario:
  - Incorrect process model: thinks spinning at less than maximum speed
    - Could be inadvertent or deliberate
- Potential controls:
  - Mechanical limiters (interlock), Analog RPM gauge

Focus on preventing hazardous state (not keeping intruders out)

How is it being used?
Does it work?
Is it useful?

Is it Practical?

- STPA has been or is being used in a large variety of industries
  - Aircraft and Spacecraft
  - Air Traffic Control
  - UAVs (RPAs)
  - Defense systems
  - Automobiles
  - Medical Devices and Hospital Safety
  - Chemical plants
  - Oil and Gas
  - Nuclear and Electric Power
  - Finance
  - Robotic Manufacturing / Workplace Safety
  - Etc.

Uses Beyond Traditional System Safety

- Quality
- Producibility (of aircraft)
- Nuclear security, nonproliferation
- Production engineering
- Banking and finance
- Engineering process optimization
- Organizational culture
- Workplace safety

Is it Effective?

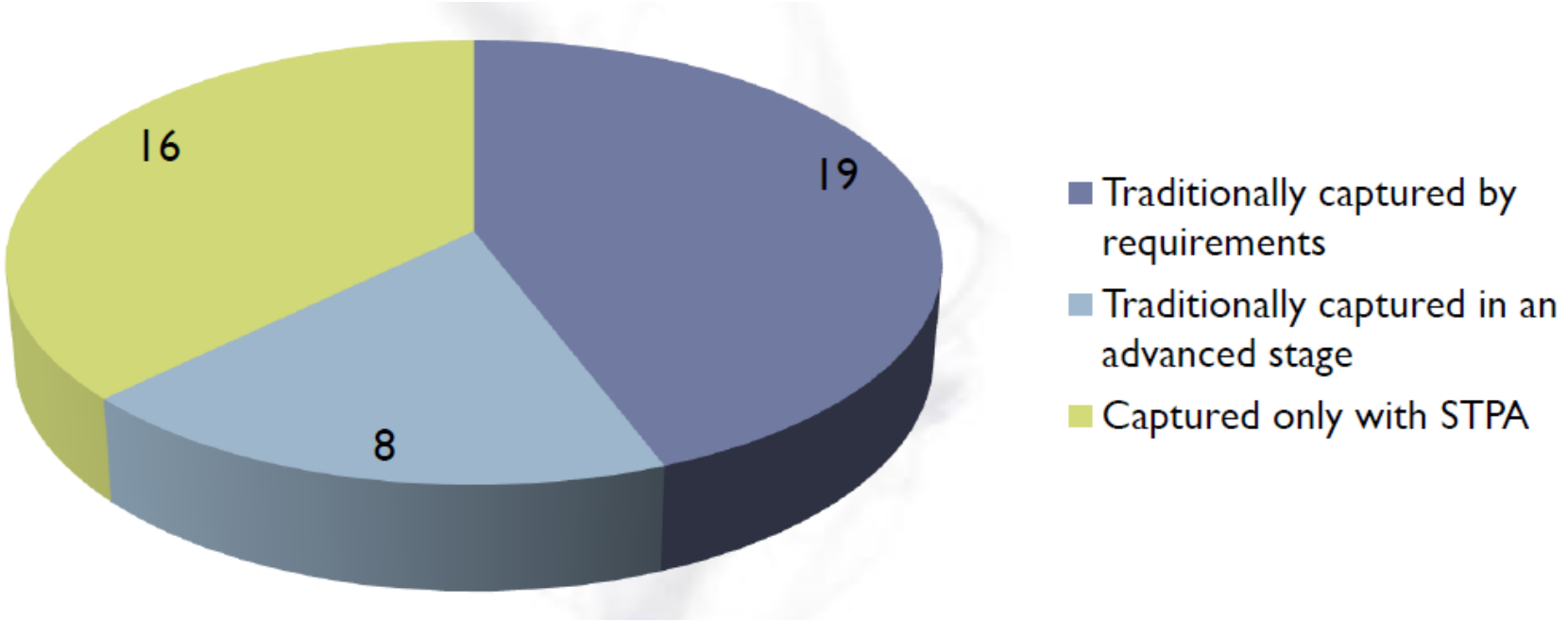
- Most of these systems are very complex (e.g., the new U.S. missile defense system)
- In all cases where a comparison was made (to FTA, HAZOP, FMEA, ETA, etc.)
  - STPA found the same hazard causes as the old methods
  - Plus it found more causes than traditional methods
  - In some evaluations, found accidents that had occurred that other methods missed (e.g., EPRI)
  - Cost was orders of magnitude less than the traditional hazard analysis methods
  - Same results for security evaluations by CYBERCOM

Some Comparisons

- EPRI Nuclear Power Plant Comparison
  - Compared FTA, FMEA, ETA, HAZOP and STPA
  - Only STPA found accident that had occurred in plant but analysts did not know about
- U.S. Navy Vessel with Dynamic Positioning System
  - Compared STPA results with official FTA/FMEA (STPA tried after 2 serious accidents during test)
  - All failures identified by FTA/FMEA identified by STPA plus lots of “non-failure” hazard causes
  - STPA identified scenarios never corrected. Put into service and collided with nuclear submarine (cause was identified by STPA)

More Comparisons

- Embraer Aircraft Smoke Control System requirements captured by STPA



- Embraer Air Management System
  - 3.5 months
  - Identified 200+ safety constraints (requirements) and 700+ design recommendations to eliminate or mitigate hazards (satisfy the safety constraints).

And More

- Blackhawk Helicopter: STPA compared with official FTA/FMEA
  - FTA/PHA identified some “hazards” as “marginal” (and thus not considered further) that STPA found led to catastrophic accidents.
  - Causal factors of FTA/FMEA limited to component failures
  - STPA identified non-failure scenarios that could lead to a hazardous state that were not identified by FTA/FMEA
  - More information about causal scenarios from STPA results led to more cost/effective mitigation measures even for failures (beyond redundancy).
  - Human error probabilities used average conditions, not worst case conditions

And Even More

- U.S. Air Force hazard analysis in flight testing vs. STPA

Traditional	STPA
2 Effects	6 Accidents
1 Test Hazard (actually a mishap)	4 System Hazards
3 Causes	392 Unsafe Control Actions
13 Minimizing Procedures *- 8 THA minimizing procedures* *- 5 general minimizing procedures*	46 Minimizing Procedures *- 14 developing influences* *- 10 settings/configurations* *- 22 operating procedures*
Nothing identified to control hazard exposure (test hazard was a mishap)	8 Corrective Actions
1 Accident-Corrective Action	7 Recovery Actions

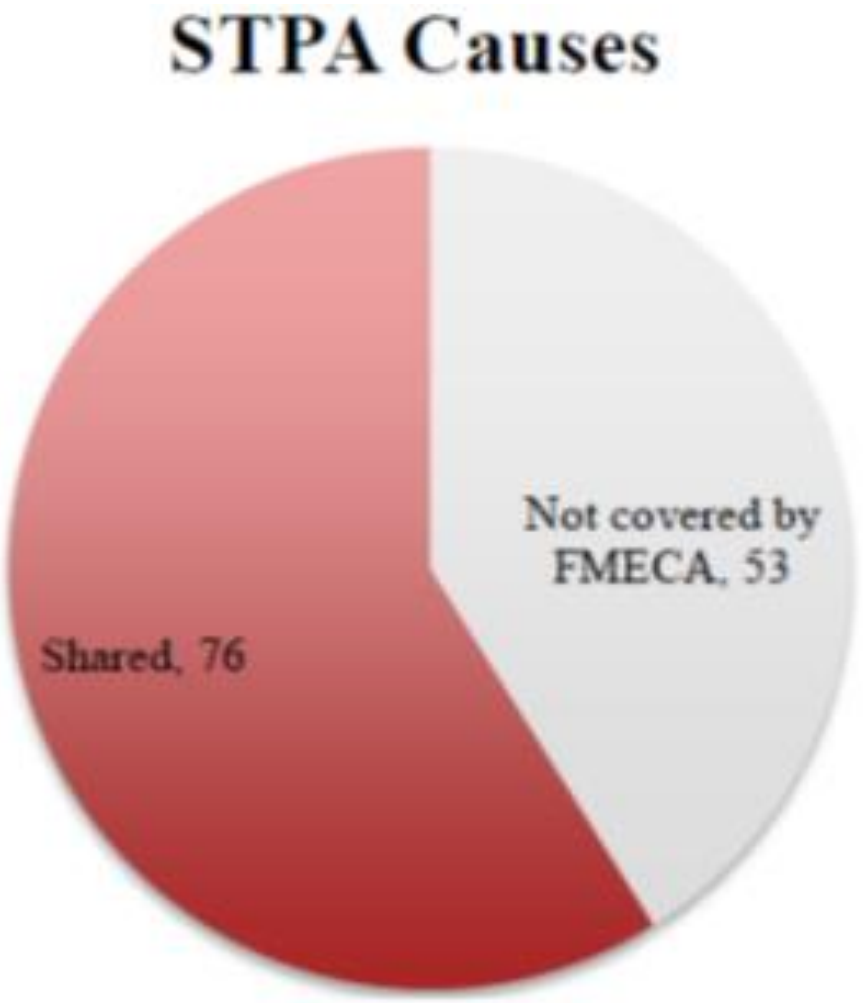
- In-Trail Procedure (NextGen/Open Skies) DO-312
  - Overlooked critical scenarios that STPA identified
  - Dismissed scenarios as “no safety effect” that STPA identified as critical
  - Human error oversimplified and superficial compared to STPA. Treated as random vs. identifying causal factors so could be reduced.

- U.S. Ballistic Missile Defense System
  - Used STPA just prior to deployment and field testing.
  - Two people, 5 months
  - Found so many paths to inadvertent launch that deployment delayed 6 months to fix them

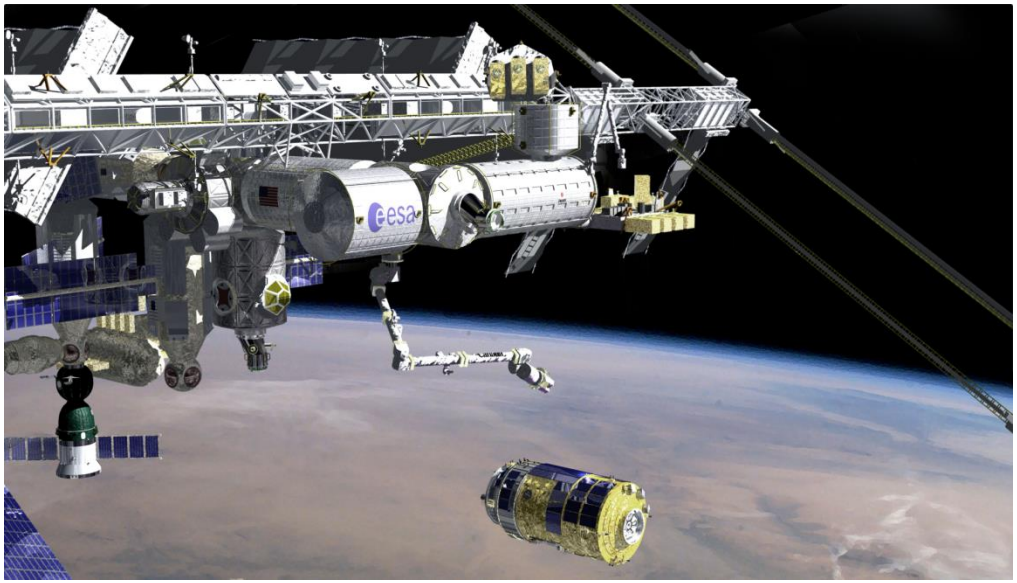
- Range Extender System for Electric Vehicles (Valeo)
  - FTA/CPA took 3 times effort of STPA, found less
- Medical Device (Class A recall)

FMECA	STPA
70+ causes of accidents	175+ causes accidents (9 related to adverse event)
Team of experts	Single semi-expert
Time dedication: months/years)	Time: weeks/month
Identified only single fault causes	Identified complex causes of accidents

- Automotive Electric Power Steering System



- HTV Unmanned Japanese Spacecraft
  - STPA found all causes found by FTA plus a lot more



Some Recent Additions to STPA

- More sophisticated human factors analysis
- Coordination between human and computer controllers (shared control)
- Organizational/managerial analysis
- Leading Indicators

Paradigm Change

- Does not imply what previously done is wrong and new approach correct
- Einstein:

  "Progress in science (moving from one paradigm to another) is like climbing a mountain"

As move further up, can see farther than on lower points

Paradigm Change (2)

New perspective does not invalidate the old one, but extends and enriches our appreciation of the valleys below

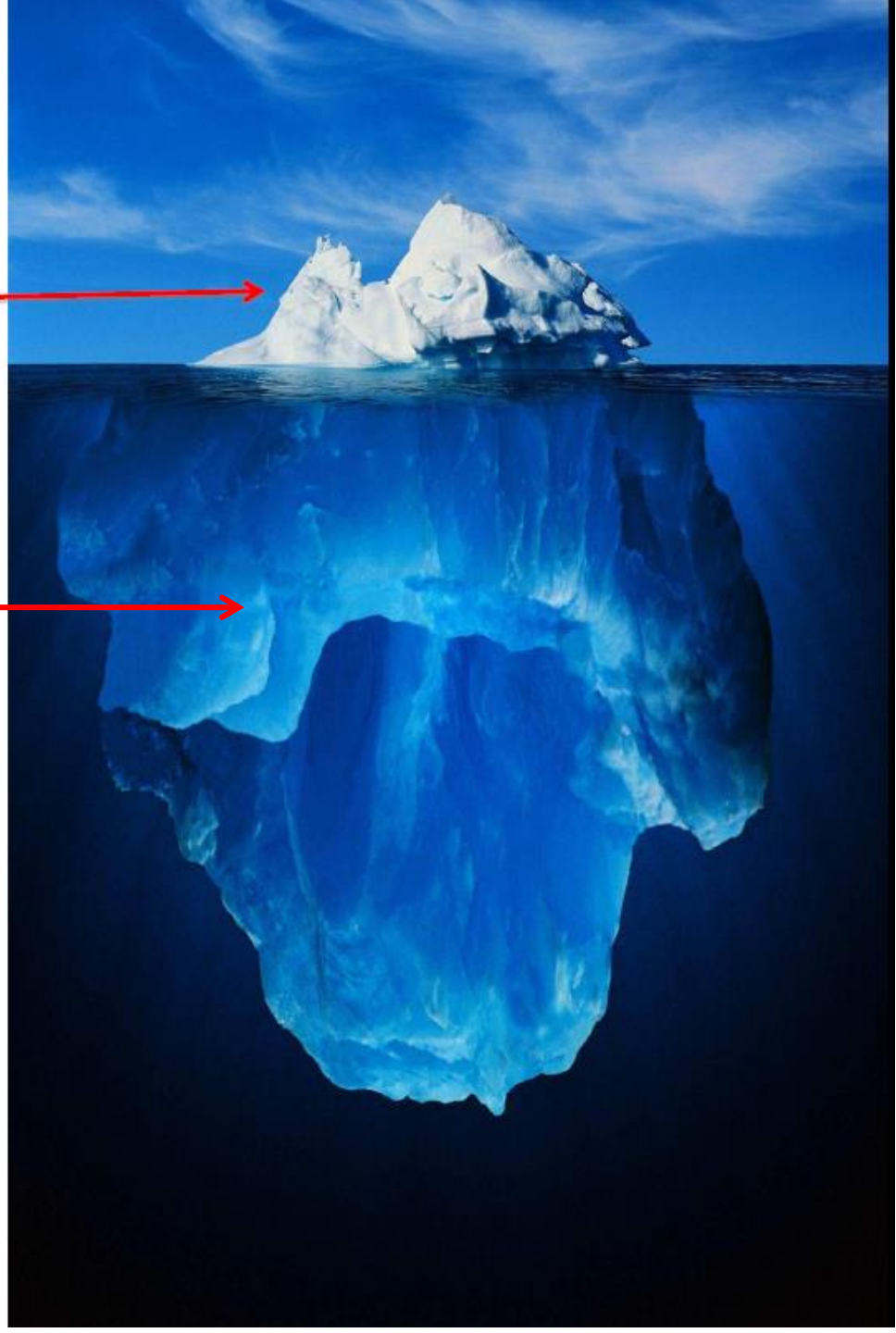
Value of new paradigm often depends on ability to accommodate successes and empirical observations made in old paradigm.

New paradigms offer a broader, rich perspective for interpreting previous answers.

LEVERAGE

Event-based thinking

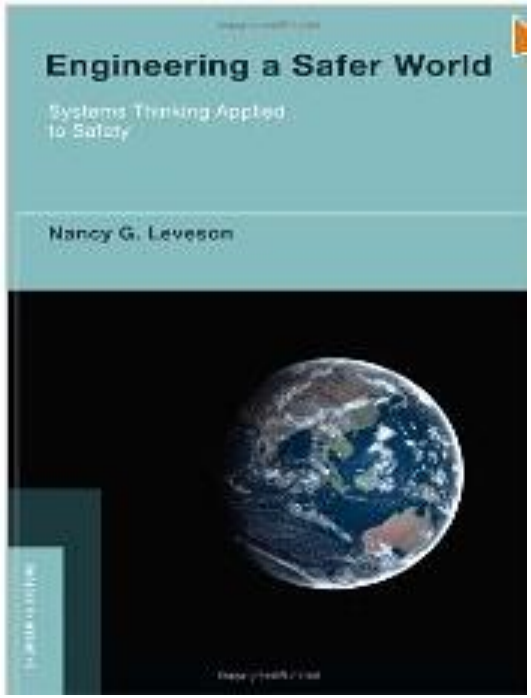
Systems Thinking



WYOUNG@MIT.EDU

Nancy Leveson, *Engineering a Safer World:*

Systems Thinking Applied to Safety



MIT Press, January 2012